United States
Naval Postgraduate School

INVENTORY POLICY EVALUATION SIMULATION

by

W. D. Free

D. A. Schrady

September 1970

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral R. W. McNitt, USN
Superintendent

M. U. Clauser
Academic Dean

ABSTRACT:

This report is a companion to the report, "Analysis of Inventory Record Accuracy" (NPS55SoFs0071A) and supplements that report by providing detailed information about the simulation model employed. The model is described, a programming guide is provided, and the program and its flow charts are given.

This task was supported by the Research and Development Division, Naval Supply Systems Command, under NAVSUP RDT&E work request WR-0-5037.

Prepared by:

TABLE OF CONTENTS

LIST OF TABLES

1. INTRODUCTION

This report is a companion to the report, "Analysis of Inventory Record Accuracy"[1], and supplements that report by providing detailed information about the simulation model employed. The simulation is useful in its own right.

The simulation models the operations of a single manager, single warehouse, multi-item inventory system under continuous review. Special attention is given to the introduction, influence, and discovery of inventory record errors. The simulation can be useful to others in the study of the effects of inventory record errors or the efficiency of various physical inventory policies. It is also useful in the evaluation of transient phenomena associated with continuous review inventory policies; for example, the quarterly procurement costs for a given set of inventory policies, and the protection or number of stockouts a given item actually experiences. In any event, the simulation should be employed to evaluate any physical inventory policy being considered for implementation within the Navy supply system.

The next section describes the model and data employed. Chapter 3 is a short programmer's guide to the use of the simulation program. The remaining four chapters present the program variable labels, the program flow charts, the program itself, and the item data employed by the authors.

---

[1]NPS55SoFs0071A, Naval Postgraduate School, Monterey, California, July 1970.

2. THE MODEL AND DATA

2.1 General Approach

The system modeled is that of a single manager, single warehouse, multi-item inventory system corresponding to a typical Navy Supply Depot activity. The simulation includes receipt and requisition processing, replenishment, and physical inventory processes. It is very nearly a general purpose inventory system model with a multitude of uses. The exact form of the model, of course, reflects a great deal of structure concerned with stock record error generation and correction. Every attempt has been made to make the simulation program, written in FORTRAN IV, as easy to understand as possible. The labels were chosen to be meaningful, so that one may view the program as the manipulation of records and quantities which are real entities in actual practice.

Two inventory options are available within the simulation: a complete wall-to-wall, with the interval between inventories as specified by the user; and a selected item inventory just prior-to-buying, which causes a scheduled* inventory to be conducted on an item just before a buy is made on that item. The length of the simulation is also selected by the user. Certain statistics are computed daily, which are used to generate annual reports. In addition, the results of each wall-to-wall inventory are printed, in conjunction with statistics which have been generated during the period since the last wall-to-wall inventory. Throughout the entire

---

*The following terminology is employed with respect to selected item physical inventories. A spot inventory is conducted in connection with a warehouse refusal. A scheduled inventory is performed in connection with a prior-to-buying physical inventory policy.

daily routine, as well as during the inventories, errors are being generated based on data presently available. At the end of the simulation, a summary of the entire run is printed, which can be used to evaluate the inventory option employed.

2.2 The Data

The item data for the original study (1) consists of information extracted from a random sample of 505 items taken ten months after a wall-to-wall inventory was conducted at NSC Newport in 1965. Of the sample of 505 items, all 187 DSA items were chosen for the present simulation. These 187 items represent a 1% random sample of the approximately 18,000 DSA items stocked at NSC Newport. The simulation employs a stock battery of 561 items which is a simple triplication of the basic 187 item sample, though the number of items in the stock battery is arbitrary.

The item characteristics selected for input to the simulation are:

- serial number, arbitrary (SERIAL)*;
- unit price (PRICE);
- recorded on-hand quantity (RECOH);
- actual on-hand quantity (ACTOH);
- dues outstanding (DUES);
- reorder quantity (Q);
- reorder point (RP);
- mean quarterly demand (DBAR);
- mean procurement lead time in quarters (LT); and
- mean absolute deviation of quarterly demand (MAD).

---

*refers to a label used in the program as the name of either a datum or variable. See Chapter 4.

NAVSUPRANDFAC (2) wall-to-wall physical inventory accuracy data is used in the simulation because of its completeness. The data indicate that for a wall-to-wall inventory, the count is correctly made 92.9% of the time. For the other 7.1% of the time, errors are assumed to be normally distributed, with a mean error quantity of zero, and a standard deviation derived from the data as follows:

ACTOH	Standard Deviation
1-10	1.87
11-20	4.65
21-100	2.20
101-	11.80

When an error is generated, the magnitude of the error is at least one unit.

Data on receipt and issue errors were obtained from the Quality Assurance and Internal Review Division of NSC Oakland. These data indicate that 95.99% of the time, the quantity ordered equals the quantity received, and the receipt is processed correctly (RECTOK). For the other 4.01% of the time, errors occur as follows:

- 1.37%: received 8% more than ordered (RECTEO)
- 1.38%: received 6% less than ordered (RECTEU)
- 0.64%: receipt not posted (RECTNP)
- 0.62%: receipt posted to wrong stock record (RECTPW).

For issue processing, the issue is correctly processed 97.74% of the time (ISSOK). For the other 2.26% of the time, errors occur as follows:

- 0.73% overissue (ISSEO)
    - 0.48% overissue by 7%
    - 0.18% overissue by 15%
    - 0.07% overissue by 30%
- 0.75% underissue (ISSEU)
    - 0.50% underissue by 8%
    - 0.12% underissue by 20%
    - 0.13% underissue by 50%
- 0.78% issue wrong item (ISSWID).

Data on scheduled inventories were also provided by NSC Oakland, and indicated that the scheduled inventory is performed correctly 96% of the time. For the other 4% of the time, errors occur as follows:

- 1%: errors are plus or minus 1
- 1%: errors are plus or minus 2
- 2%: errors vary from 5 to 100, as a function of ACTOH.

For spot inventories the assumption is made that an accurate reconciliation is made 97% of the time and that the record remains unchanged 3% of the time. This is a crude assumption, but fortunately the spot inventory plays a very minor role in the simulated inventory operations.

Data on inventory costs were provided by FMSO (3). For conducting a wall-to-wall inventory, the cost is estimated to be $1.09 per item. For conducting a selected item inventory, the cost is estimated to be $3.85. A spot inventory was estimated to cost $3.92. These costs include reconciliation, in addition to the physical counting process.

2.3 Underlying Distributions

DBAR and MAD item demand data from FMSO records is used to specify the parameters of the item demand generation processes. Realistic simulation requires that requisitions be generated over time in some fashion, and that the total number of items requisitioned per quarter approximate DBAR. Both the time between successive requisitions on an item and the size of the requisitions must be specified. To assume that all requisitions are for a quantity of one would be misleading since the total number of requisitions the system would have to process would be too large. Since each requisition and the attempted issue which follows from the requisition can generate errors, it is important that a variable requisition size be used.

The assumption made in this study is that item demand follows a "stuttering Poisson" distribution; that is, that the time between successive demands is exponentially distributed and that the requisition size has a geometric distribution on the positive integers. The method of moments is used to estimate the parameters of the stuttering Poisson from DBAR and MAD for each item. Equations for these estimates are developed in Appendix B of (1).

The daily operations of a stock point are dynamic, and any model which attempts to simulate such operations must provide the randomness which is needed. In the above, the underlying distributions for the simulation were stated, with no indication of how these distributions were to be generated. In order to provide the randomness called for, and to meet the basic criteria of the distributions as stated above, pseudo random numbers are generated which are inputs to subroutines which output random variables with various distributions as required. The stream of numbers is random in that it meets certain statistical tests for randomness, and is pseudo

in that any given stream of random numbers can be reproduced. In the present simulation, the IBM subprogram RANDU is used to generate three different streams of random numbers: one for generating demands, one for generating errors of various types, and one for all other uses. These three different streams allow comparisons to be made between different inventory policies. It is assumed that a valid comparison of inventory policies is possible only when the stock sample being used faces the same pattern of demand from run to run. This scheme also provides for allowing runs to be made with no errors at all, these "clean" runs providing a benchmark for the "dirty" runs (those runs in which errors are introduced).

2.4 The Vector Framework

The basic framework of the simulation is two vectors: a stock record or item vector, and a buy vector. An item vector has various components which allow the item to be identified, and which allow the status of the item to be maintained. The components of each item vector are: SERIAL, PRICE, RECOH, ACTOH, DUES, Q, RP, DBAR, LT, and MAD, as defined in Section 2.2 and, in addition, the following:

- quantity presently backordered (BO);
- total dollar value of buys to date (CUMBUY);
- demand parameters (P, NU);
- date of next requisition (NXTREQ);
- total cumulative demand to date (CUMDMD); and
- total cumulative backorder days to date for this year (BODAYS).

The subscript I is employed throughout the simulation to refer to the Ith item; for example, if I = 25 , then PRICE(I) is the unit price of item number 25.

The second vector is the buy vector, which consists of the following components:

- the item number to which the buy applies (INDEX);
- the quantity ordered (ORDQN); and
- the due date of the material (DUEDAT).

The subscript J is employed throughout the simulation to refer to the Jth buy; for example, if J = 681 , then INDEX(J) holds the item number for which the 681st buy was made, e.g., item number 25. The use of these vectors will be clarified in the following paragraphs.

2.5 Program Operations and Control

The reader's attention is invited to the system flow chart, as well as the detailed flow charts. The present discussion will provide a narrative clarification of these charts, and is intended to provide an appreciation of the logical construction of the model.

After the standard initialization procedures, the item data is read, after which the user, assumed to be controlling the program through a time-sharing terminal, is queried for the length of the simulation, the type of run (clean or dirty), the demand random number streams initializer (INITRN), and the inventory option desired. The value of INITRN is arbitrary and merely provides for a constant demand pattern from run to run

if desired. The simulation is now independent of the user, and proceeds as follows. Due dates are computed for any outstanding dues at the beginning of the simulation, by assuming that all material will arrive during the first sixty days of the run in accordance with a uniform distribution. Then the parameters of the stuttering Poisson demand generators are computed from item DBAR and MAD data. Dates of first requisitions for each item (NXTREQ) are computed using the exponential distribution, and all records are scanned to determine if buys are necessary. (Conceptually, the generation of dates for the next requisition to occur is a type of 'event-store' process, in which the occurrence of a particular type of transaction causes the generation of the time for the next transaction of the same type to occur.)

The decision as to whether to make a buy is made by the subroutine BUY by computing the inventory position (IP) of the item, defined to be the recorded on-hand quantity (RECOH) plus outstanding dues (DUES) minus backorders (BO). If the inventory position is less than or equal to the reorder point (RP), an order is generated for the integer multiple of the order quantity (Q) which will bring the inventory position up to a point between RP and RP + Q.

Having completed the initialization procedures for day one, the daily routine begins. Each day is identified by an integer number; the present day at any time is the value of the variable TODAY. Thus the simulation begins with TODAY equal to one, and time-steps through to TODAY equal to FINISH, which is the last day of the simulation.

The daily routine begins with receipt processing, which consists of scanning the list of outstanding orders to see if any dates in the DUEDAT

vector match the date in TODAY. If there is a match for a particular item, a random number if generated to determine whether the receipt will be processed correctly, or with errors. After the receipt is processed, a check is made to determine if there are any backorders outstanding for that item. If so, an attempt is made to release the backorders. (Note: even though requisitions of different sizes are generated, only the total quantity of each requisition backordered is recorded in BO. No provision is made to distinguish a backorder resulting from one requisition from that resulting from any other requisition. Thus, the simulation can only keep statistics on the number of requisitions which have been backordered.) Backorder releases are very similar to regular issues, in which the record is checked to determine if an attempted issue should be made, after which the actual on-hand quantity is checked in order to actually effect the issue. In the case of a backorder release, an attempted issue quantity (ISSQN) assumes the value of the present number of backorders (BO), and a determination if made of whether an actual issue can be made. If so, the subroutine ISBOER is called to make the backorder release subject to errors.

After each issue of material on an item, whether to release a backorder or to satisfy a requisition, a check of the new inventory position is made. If the new inventory position is at or below the reorder point, a buy is made in order-quantity multiples to bring the inventory position back up to the (RP, RP + Q) interval.

After the receipt and backorder processing is completed, all requisitions are processed. The component NXTREQ is checked against TODAY to determine if there is to be a requisition today for the particular item in

question. If so, the requisition size (REQSIZ) is generated using the geometric distribution, and a new NXTREQ is generated for the item from its exponential inter-demand time distribution. As in the backorder release routine, the attempted issue quantity (ISSQN) is determined by REQSIZ, and both the recorded on-hand quantity (RECOH) and the actual on-hand quantity (ACTOH) are checked to see if the issue can be made. If necessary, a warehouse refusal is generated which results in a spot inventory being taken, subject to errors. The actual issue is made by calling the subroutine for making issues with errors (ISSERR), the flowchart of which is not included since it is so similar to ISBOER.

The end of the day brings the daily update to keep track of such items as total accumulated unit backorder days to date this year (BODAYS), the record accuracy at the end of the day, and the dollar value of investment recorded and actually held on hand. If today is the day for a wall-to-wall inventory, the subroutine WALLOP is called. If today is the end of a year, the annual report is generated. If today is the end of the quarter, the subroutine QTR$ is called to compute the dollar value of demand for this quarter and the dollar value of buys for this quarter. If today is the end of the simulation, the summary is generated. If today is not the last day, the variable TODAY is incremented, and the daily routine begins again.

2.6 The Output

The original purpose of the simulation was to determine the effects of record accuracy upon measures of effectiveness and costs for the system. Accordingly, the output of the simulation was designed to allow comparisons over time of certain statistics, as well as to allow evaluation of the entire run. Table 1 is a typical annual report.

TABLE 1. TYPICAL ANNUAL REPORT

SUPPLY PERFORMANCE MEASURES:

CUMREQ = 2791
COMPFL = 2454 = 87.93 Per Cent of CUMREQ
PARTFL = 316 = 11.32
ACOMFL = 2487 = 89.11
APARFL = 304 = 10.89
ABOREL = 123
BOREL = 123
TOTAL BODAYS (EODTOT) = 661227
BUYS = 722
REFUSL = 7

ERROR MEASURES:

RECTOK = 694 = 95.33 Per Cent of Total Receipts
RECTEO = 14 = 1.92
RECTEU = 10 = 1.37
RECTNP = 5 = 0.69
RECTPW = 5 = 0.69
ISSOK = 2850 = 98.04 Per Cent of total Issues (Includes BO releases)
ISSEO = 22 = 0.76
ISSEU = 15 = 0.52
ISSWID = 20 = 0.65

QTR	DEMAND$	BUY$
5	20710.69	20811.81
6	20371.13	19452.00
7	15642.81	15522.56
8	1''774.37	17279.88

Each measure applies only to the year immediately preceding the day of the report. The measures are defined as follows:

- CUMREQ: total number of requisitions;
- COMPFL: number of requisitions completely filled on demand;
- PARTFL: number of requisitions partially filled on demand;
- ACOMFL: number of attempted complete fills;
- APARFL: number of attempted partial fills (the difference which may result in actual versus attempted is due to the effects of record inaccuracies);
- ABOREL: number of attempted backorder releases;
- BOREL: number of actual backorder releases;
- TOTAL BODAYS (BODTOT): total unit backorder days, in millions;
- BUYS: number of buys; and
- REFUSL: number of warehouse refusals.

The error measures are the same as those stated above in Section 2.2, and represent the mean accuracies of the stochastic processes generating receipt and issue errors. DEMAND$ and BUY$ provide the dollar values of quarterly demand and buys for the quarter indicated.

When the wall-to-wall inventory option is selected, a report of the inventory results and statistics accumulated since the last wall-to-wall inventory are printed. Table 2 is an example of such a report.

TABLE 2. TYPICAL WALL-TO-WALL INVENTORY REPORT

728 = DAY WALL-TO-WALL INVENTORY HELD

89.48 = PER CENT RECORDS ACCURATE JUST PRIOR TO INVENTORY

90.82 = MEAN PER CENT RECORD ACCURACY DURING PERIOD

88.77 = MINIMUM PER CENT RECORD ACCURACY DURING PERIOD

50456.00 = MEAN DOLLAR VALUE OF RECOH DURING PERIOD

45360.80 = MINIMUM DOLLAR VALUE OF RECOH DURING PERIOD

50372.13 = MEAN DOLLAR VALUE OF ACTOH DURING PERIOD

45386.03 = MINIMUM DOLLAR VALUE OF ACTOH DURING PERIOD

91.80 = PER CENT RECORDS ACCURATE JUST AFTER INVENTORY

Information from both of the above reports is accumulated for the summary at the end of the simulation. The summary consists of two parts, one of which summarizes information collected on an annual basis, the other of which summarizes information related to the wall-to-wall inventory periods. If the PBUY inventory option is selected, the periodic portion of the report contains information accumulated annually. Table 3 is a typical summary.

The annual portion of the summary is derived from the individual annual reports which are generated as the simulation proceeds. DIFF is the difference between COMPFL and PARTFL, and represents those requisitions which resulted in a backorder for the full amount of the requisition (a 'no-fill'). The periodic portion of the report shows the inventory period by number, the mean record accuracy for the period (RECACC), and the mean dollar value of investment during the period, both recorded (RECOH$)

TABLE 3. TYPICAL SUMMARY REPORT

SUMMARY STATISTICS

ANNUAL:

YEAR	COMPFL NO.	PC	PARTFL NO.	PC	DIFF NO.	PC	BOREL	BODAYS	BUYS	REFUSL
1	2583	92.9	167	6.0	29	1.0	70	109269	904	11
2	2454	87.9	316	11.3	21	0.8	123	661227	722	7
3	2446	86.1	379	13.3	17	0.6	180	1147651	771	8
4	2460	88.1	308	11.0	24	0.9	135	835196	742	10
5	2320	83.4	442	15.9	20	0.7	179	1195012	748	9
6	2395	83.7	440	15.4	27	0.9	176	976435	782	10
7	2435	84.9	413	14.4	21	0.7	147	1046662	746	7
8	2229	83.7	415	15.6	20	0.8	164	620770	703	11
MEAN	2391	85.4	387	13.8	21	0.8	157	926136	744	8

PERIODIC:

PERIOD	RECACC	RECOH$	ACTOH$
1	93.7	57047.07	57077.49
2	90.8	50456.09	50372.13
3	88.6	49017.62	49021.89
4	92.9	48331.78	48077.89
5	92.2	49379.91	49006.15
6	90.4	46168.09	46159.95
7	90.1	49586.15	45575.49
8	92.2	46527.82	45575.49
SUM	730.9	396534.37	395272.81

and actual (ACTOH$). This latter value of mean actual investment represents information which is never actually available to the stock point manager, and which will be used to cost out the effects of errors in the system. The row labelled SUM can be used to average RECACC, RECOH$, and ACTOH$ for any number of periods desired.

3. PROGRAMMING GUIDE

In addition to the description of the model, it is deemed necessary to further assist the potential user with some amplifying remarks. Some of the points would be obvious with detailed study of the program, other points would not become obvious without some experience in running the program.

1. The item input data consists of one item per card, formatted as follows:

cc	CHARACTERISTIC	FORMAT
1- 3	SERIAL	I3
22-26	PRICE	F5.2
28-31	RECOH	I4
33-36	ACTOH	I4
38-40	DUES	I3
42-45	Q	I4
47-50	RP	I4
52-57	DBAR	F6.2
59-60	LT	F2.1
62-66	MAD	F5.2

The serial number if arbitrary; there is enough space between SERIAL and PRICE to insert a stock number, if desired.

2. The simulation is currently set up to use 561 stock records. As such, the core required in the IBM 360/67 is approximately 230K bytes. This storage requirement obviously depends upon the number of stock records used. In addition, it is quite dependent upon the activity of the items in that the buy vectors (INDEX, ORDQN, and DUEDAT) are required to

be longer for a more active item sample, since the number of buys per year will be greater for more active items than for less active ones.

3. The average cpu time per year of simulation is about one minute on the 360/67, if 3-4 years or more are run. Actually, the time required per year of simulation increases from year to year, due to the search procedure employed for receipt processing, i.e., each day, the entire DUEDAT vector is searched to determine the receipts due in. As noted above, the DUEDAT vector becomes longer each time a buy is generated.

4. The simulation files are set up as follows:

DATASET NUMBER	FILE NAME
4	Item input data
5	Terminal input data
6	Terminal output
8	Printer output

Obviously, these files can be set up to suit the individual user. If the program is to be run as a batch job, the input from the terminal can be modified for card input.

5. Following is an example of the terminal I/O:

```
EXECUTION BEGINS...
PRESENT PARAMETERS ARE:
& INPUT
INITRN=        100,RUN=          1,WALINT=  364.00000
& END
ANY PARAMETER CHANGES?
n
DURATION IN YEARS OR DAYS?
y
DURATION?
2.
INVENTORY OPTION:  WALL OR PBUY?
wall
INVENTORY INTERVAL IN DAYS?
182.
```

The values of INITRN, RUN, and WALINT are preset, and may be left unchanged if desired. A value of 1 for RUN produces a dirty run: a value of 0 produces a clean run. WALINT is set to 364, in case the PBUY option is selected, so that annual results will be printed. The duration may be specified in either years or days. If the duration is less than one year, an abort will occur when the program reaches the point at which the summary statistics are to be printed, since the annual portion of the summary statistics does not include the first year in the computation of the means. The option of selecting the duration in days may be helpful when the present status of each stock record at the end of the simulation is desired. For example, one might wish to see specifically what occurred on day 722. He would then run the simulation for 721 days, and subsequently, for 722 days, electing to have the stock battery vectors printed at the end of each run.

6. E8 is a local subroutine which allows efficient dumping of output periodically onto the output spool under CP/CMS, which is the present time-sharing package under which the simulation has been run. The purpose of E8 is to avoid filling up the user disk file with output, thus causing an abort in the program. The user should be aware that the simulation can produce a great quantity of output if desired, and should thus ensure that sufficient precautions are taken to handle the output.

7. The user will notice that in many arithmetic statements, .5 is added to the right hand side. This produces an upward rounding off before truncation, since many quantities dealt with in the simulation are small integer numbers.

It is apparent that the program will require some modifications prior to being run on any machine other than that for which it was designed. For example, the instruction to call E8 must be deleted. The dataset numbers will probably have to be changed, etc. The authors assume that the user is adept at programming, however, and feel that the changes which will be required are nominal.

4. SIMULATION PROGRAM LABELS

ABOREL:	Number of attempted backorder releases for year.
ACOMFL:	Number of attempted completely filled requisitions for year.
APARFL:	Number of attempted partially filled requisitions for year.
ACTOH(I):	Actual on-hand quantity for Ith item.
ACTOH$(I):	Dollar value of actual on-hand inventory for the Ith day.
BO(I):	Amount of material backordered for Ith item.
BODAYS(I):	Number of backorder days for Ith item for year.
BOREL:	Number of actual backorder releases for year.
BODTOT:	Total unit backorder days at end of year for that year.
BUY:	Subroutine for checking inventory position on an item, and initiating a buy if necessary.
BUYS:	Number of buys for a year.
BUY$(I):	Dollar value of buys for Ith quarter.
COMPFL:	Number of actual completely filled requisitions for year.
CUMREQ:	Number of requisitions for year.
DBAR(I):	Mean quarterly demand on Ith item (data).
DUEDAT(I):	Due date of material ordered on Ith buy for the item number contained in INDEX(I).
DUES(I):	Amount of material due in for the Ith item.
FINISH:	Day on which simulation terminates.
IFLAG:	Flag generated during requisition processing which determines for ISSERR subroutine whether ISSQN should be COMPFL or a PARTFL.
INDEX(I):	Contains item number of Ith buy.
INVOP:	Inventory option (WALL of PBUY).
ISBOER:	Subroutine for making backorder releases with errors.

ISSERR: Subroutine for making requisition issue with errors.

ISSOK: Number of requisitions issued without error for year.

ISSEO: Number of requisitions overissued for year.

ISSEU: Number of requisitions underissued for year.

ISSWID: Number of requisitions in which wrong stock number was issued for year.

ISSQN: Issue quantity; used in both issue and backorder release subroutines.

LT(I): Mean procurement lead time for Ith item (data).

MAD(I): Mean absolute deviation of quarterly demand for Ith item (data).

NEWREC: Item number of new record chosen at random by subroutine NUREC.

NU(I): Parameter for exponential demand distribution for Ith item (computed from data).

NUREC: Subroutine for randomly selecting record in same price range as record under consideration.

NXTREQ(I): Day of next requisition on Ith item (recomputed every time a requisition is received on Ith item).

ORDQN(I): Quantity of material ordered on Ith buy for item number contained in INDEX(I).

P(I): Parameter of geometric distribution employed in determination of requisition size (REQSIZ) on Ith item (computed from data).

PARTFL: Number of actual partially filled requisitions for year.

PBUY: Prior-to-buy inventory option.

PRICE(I): Unit price of Ith item (data).

RECACC(I): Proportion of records accurate at end of Ith day.

RECOH(I): Recorded on-hand quantity for Ith item.

RECOH$(I): Dollar value of recorded on-hand inventory for the Ith day.

RECTOK: Number of receipts processed without error for year.

RECTEO: Number of receipts for year with quantity actually received greater than quantity ordered.

RECTEU: Number of receipts processed for year with quantity actually received less than quantity ordered.

RECTNP: Number of receipts for year with no posting to recorded on-hand and dues fields of records.

RECTPW: Number of receipts for year with quantity posted to recorded on-hand field of randomly selected record, using NUREC subroutine.

REFUSL: Number of warehouse refusals for year.

REQSIZ: Requisition quantity generated from geometric distribution upon receipt of a requisition.

RP(I): Reorder point for Ith item (data).

SERIAL(I): Stock (item) number (sequential from 1, 2, ...) of Ith item.

SPOT: Subroutine for conducting spot inventory, called every time a warehouse refusal is generated, or with PBUY physical inventory.

TODAY: Current date (integer number, begins with 1).

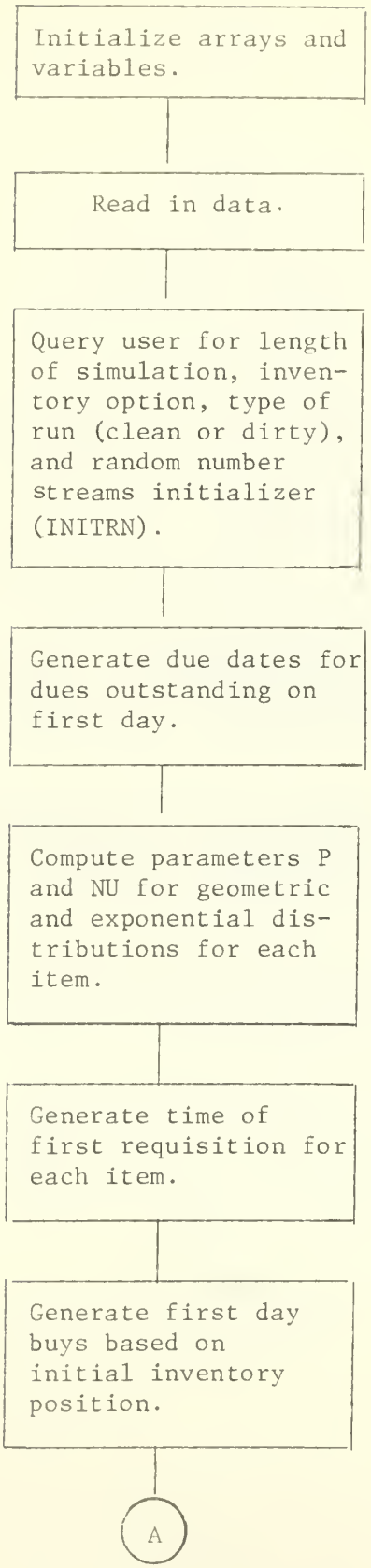
WALL: Wall-to-wall inventory option.

WALINT: Interval between wall-to-wall inventories; provided by user.

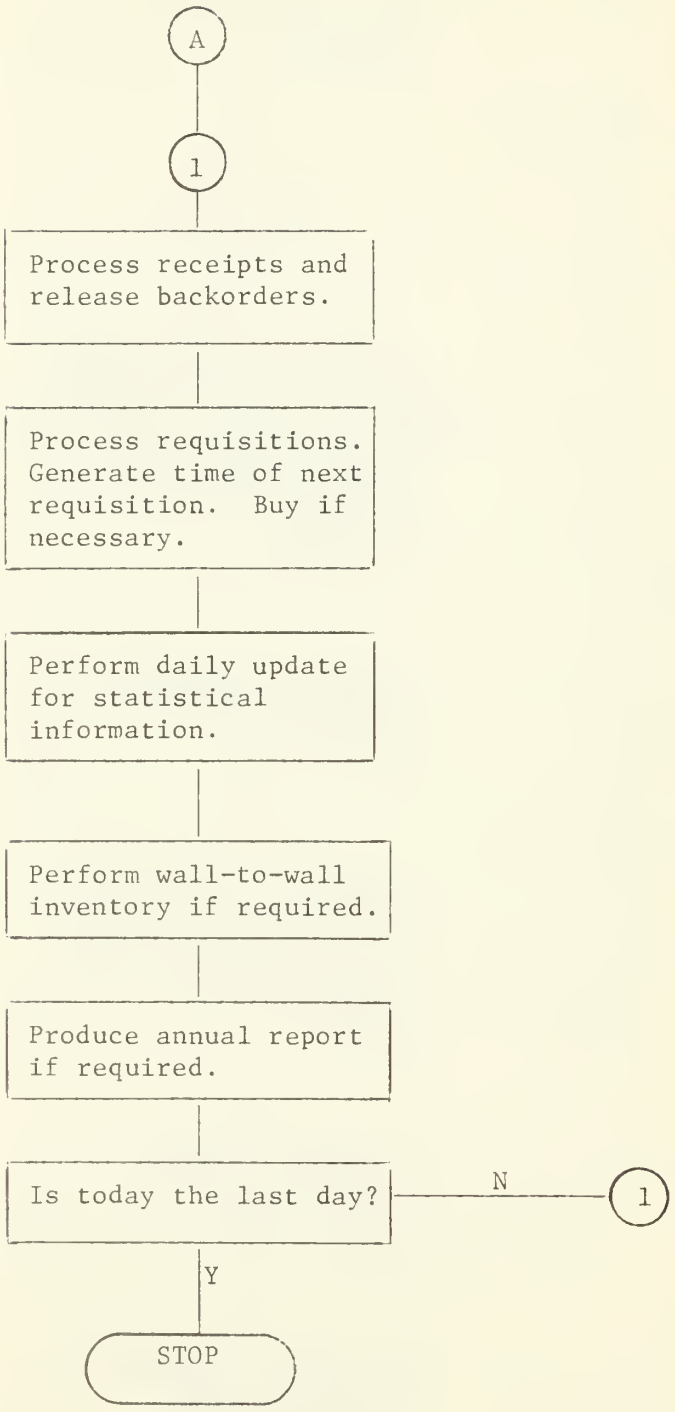
WALLOP: Subroutine for conducting wall-to-wall inventories and producing certain statistics pertinant to the period just preceding the inventory.

5. FLOW CHARTS FOR SELECTED PORTIONS OF THE SIMULATION PROGRAM

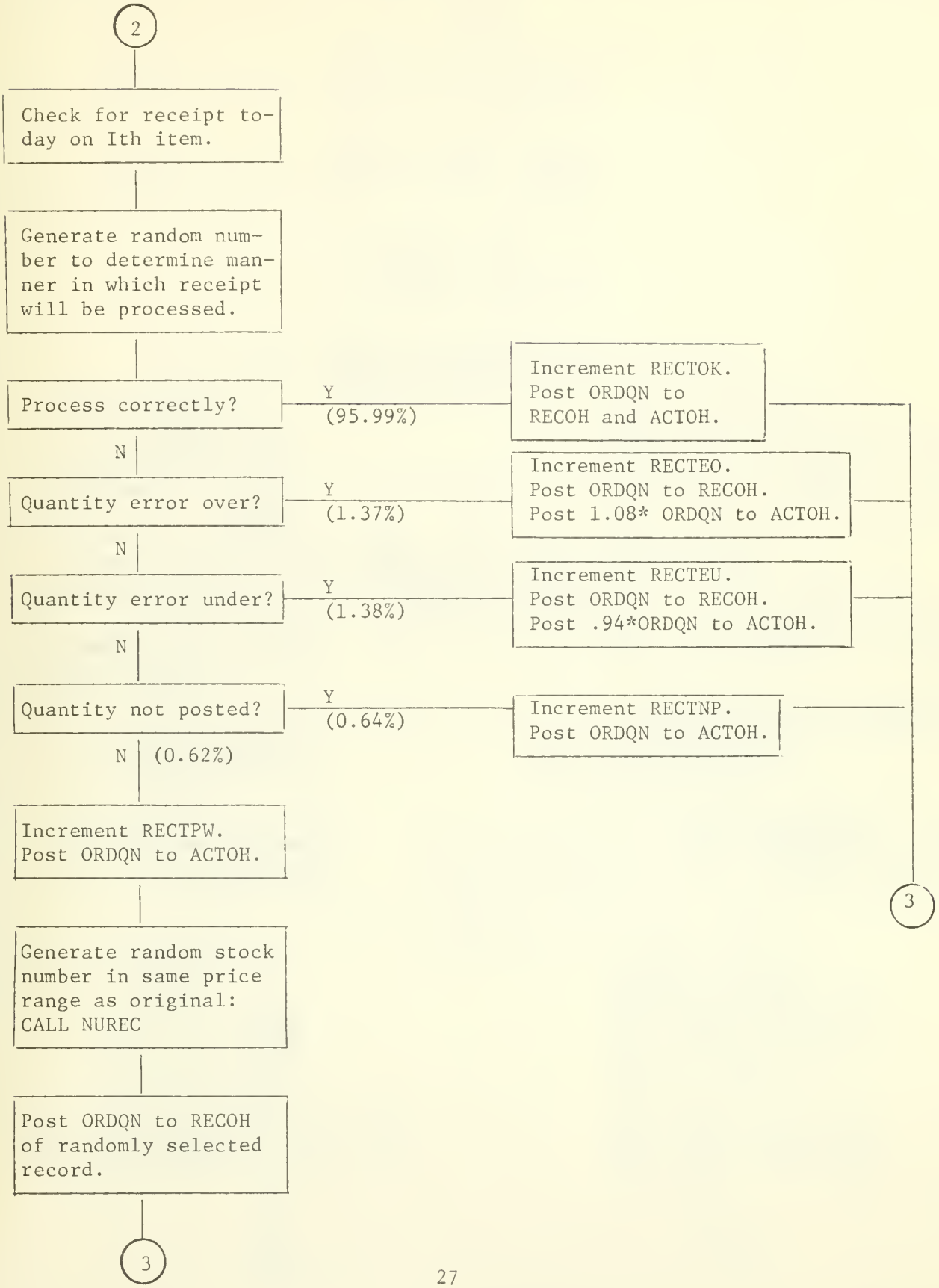
SYSTEM FLOW CHART



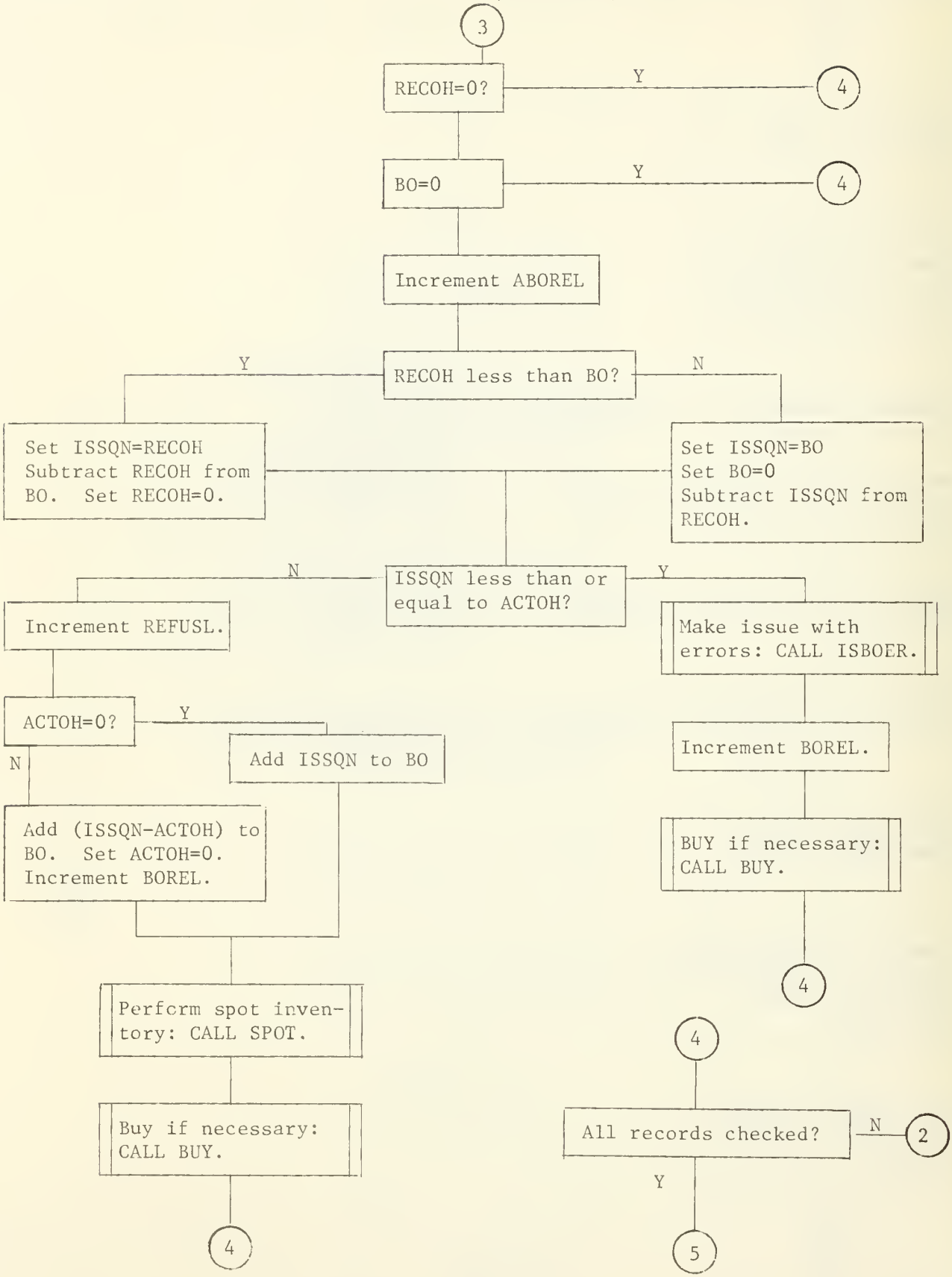
SYSTEM FLOW CHART (continued)

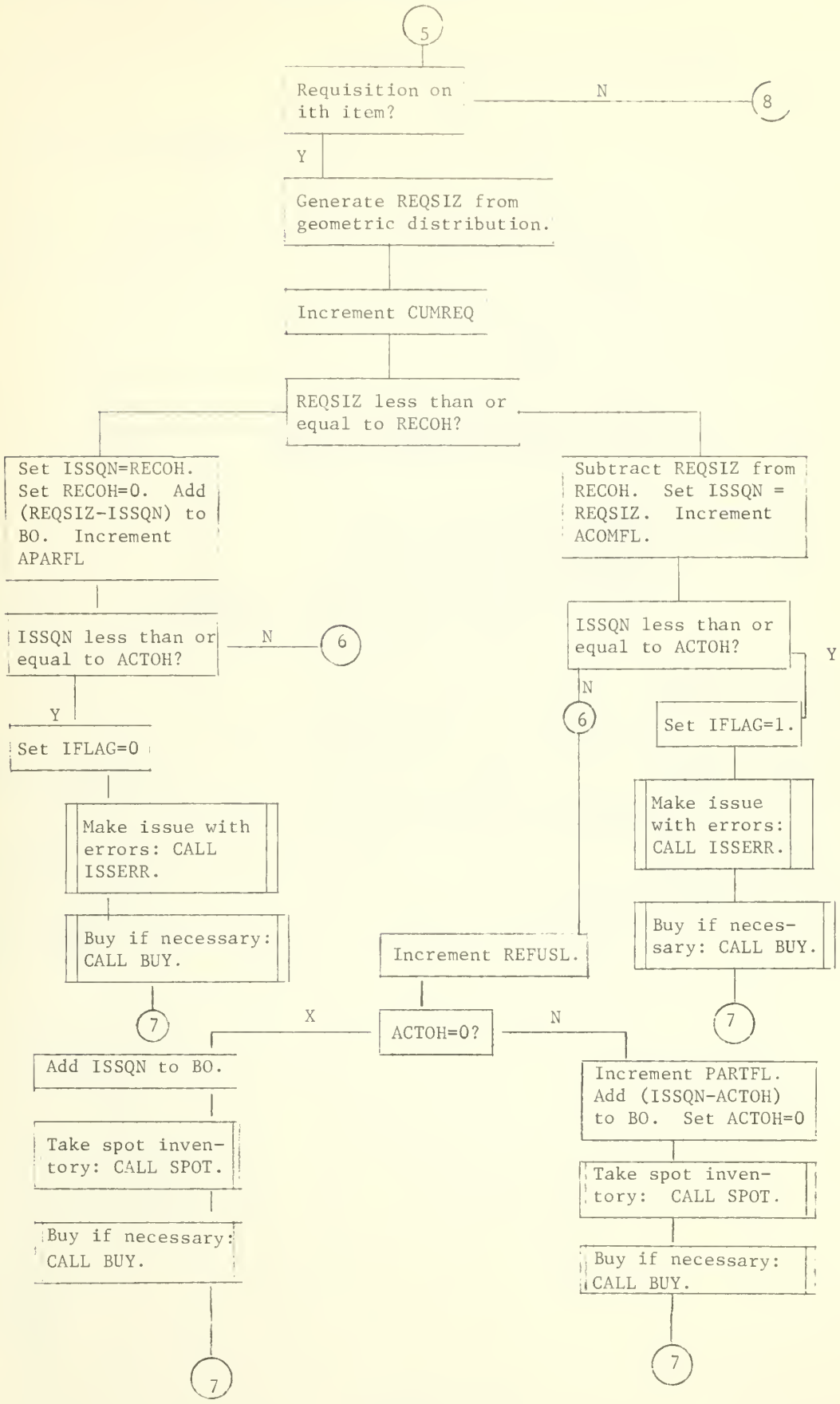


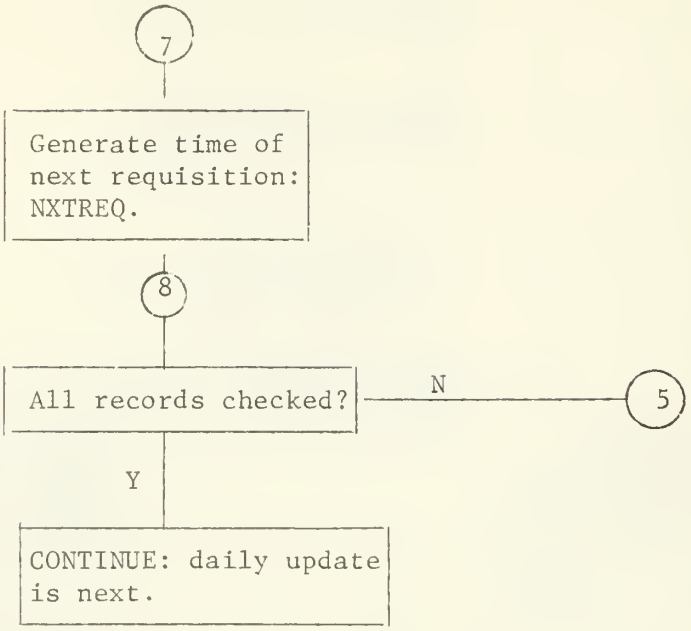
PROCESS RECEIPTS AND RELEASE BACKORDERS



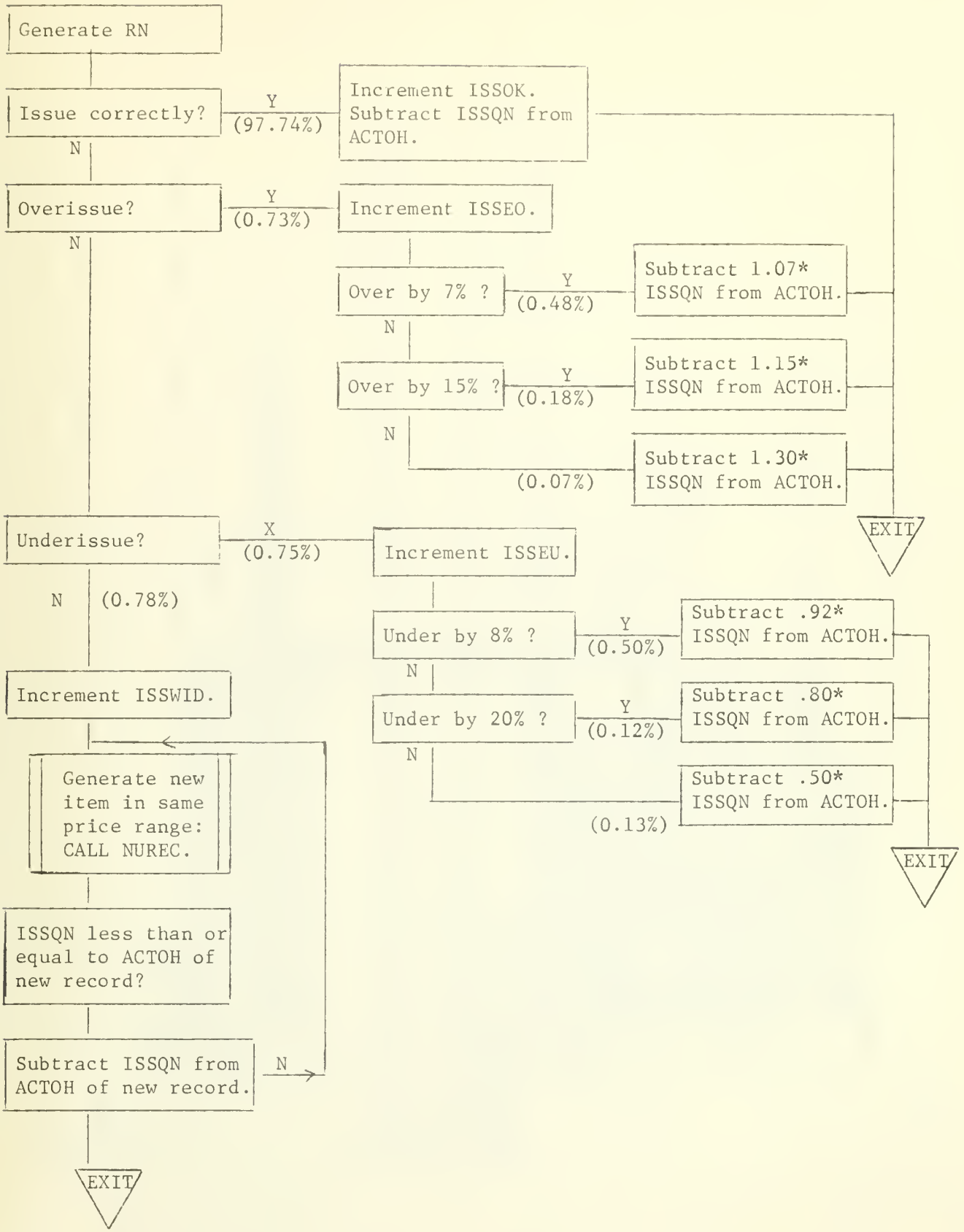
PROCESS RECEIPTS AND RELEASE BACKORDERS (continued)




5
Requisition on ith item?
N
8
Y
Generate REQSIZ from geometric distribution.
Increment CUMREQ
REQSIZ less than or equal to RECOH?
Set ISSQN=RECOH. Set RECOH=0. Add (REQSIZ-ISSQN) to BO. Increment APARFL
ISSQN less than or equal to ACTOH?
N
6
Y
Set IFLAG=0
Make issue with errors: CALL ISSERR.
Buy if necessary: CALL BUY.
7
Subtract REQSIZ from RECOH. Set ISSQN = REQSIZ. Increment ACOMFL.
ISSQN less than or equal to ACTOH?
Y
N
6
Set IFLAG=1.
Make issue with errors: CALL ISSERR.
Buy if necessary: CALL BUY.
7
Increment REFUSL.
ACTOH=0?
X
N
Add ISSQN to BO.
Take spot inventory: CALL SPOT.
Buy if necessary: CALL BUY.
7
Increment PARTFL. Add (ISSQN-ACTOH) to BO. Set ACTOH=0
Take spot inventory: CALL SPOT.
Buy if necessary: CALL BUY.
7


7
Generate time of next requisition: NXTREQ.
8
All records checked?
N
5
Y
CONTINUE: daily update is next.

SUBROUTINE ISBOER: Issue (release) backorders with errors.



6. THE FORTRAN IV SIMULATION PROGRAM

```
C.....................................................................
CCC
CCC BLOCK DATA SUBPROGRAM.  CAN BE PLACED AT END OF PROGRAM IF DESIRED.
C
      BLOCK DATA
      COMMON/BUYY/ RECOH,DUES,BO,RP,Q,INDEX, DUEDAT,ORDQN,LT,TODAY,BUYS,
     X   CUMBUY,ACTOH,RUN,INVOP,PBUY
      COMMON /ISSUE/ ISSOK,ISSEO,ISSEU,ISSWID
      REAL LT(561),INVOP,PBUY/'PBUY'/
      INTEGER ISSOK/0/,ISSEO/0/,ISSEU/0/,ISSWID/0/,RECOH(561),
     X   DUES(561),BO(561),RP*2(561),Q*2(561),INDEX*2(6500),
     X   DUEDAT*2(6500),ORDQN*2(6500),TODAY/1/,BUYS/0/,
     X   CUMBUY(561),ACTOH(561),RUN/1/
      END
CCC.....................................................................
CCC.....................................................................
CCC.....................................................................
CCC
CCC MAIN PROGRAM BEGINS HERE.
CCC
CCC THROUGHOUT PROGRAM, .5 ADDED ON RIGHT SIDE OF STATEMENT PROVIDES
CCC ROUNDING FEATURE.
C
      COMMON /BUYY/ RECOH,DUES,BO,RP,Q,INDEX,DUEDAT,ORDQN,LT,TODAY,BUYS,
     X   CUMBUY,ACTOH,RUN,INVOP,PBUY
      COMMON /ISSUE/ ISSOK,ISSEO,ISSEU,ISSWID
      REAL PRICE(561),LT(561),
     X   CUMDM$(80),BUY$(80),
     X   DBAR(561),MAD(561),P(561),NU(561),
     X   WALL/'WALL'/,PBUY,TRIG/'TRIG'/,INVOP,Y/'Y   '/,
     X   D/'D   '/,WALINT/364./,
     X   RECOH$(3000),ACTOH$(3000),
     X   RECACC(3000),ISSOKP,ISSEOP,ISSEUP,ISWIDP,
     X   CUMISS,CUMRCT
      INTEGER SERIAL*2(561),ACTOH(561),
     X   NXTREQ*2(561),CUMDMD(561),
     X   BODAYS(561),
     X   FINISH,RECTOK/0/,RECTEO/0/,
     X   RECTEU/0/,RECTNP/0/,RECTPW/0/,ABOREL/0/,BOREL/0/,
     X   REFUSL/0/,REQSIZ,CUMREQ/0/,
     X   ACOMFL/0/,APARFL/0/,
     X   COMPFL/0/,PARTFL/0/
      INTEGER RECOH(561),DUES(561),BO(561),RP*2(561),Q*2(561),
```

```
     X    INDEX*2(6500),DUEDAT*2(6500),ORDQN*2(6500),TODAY,BUYS,
     X    CUMBUY(561),QTRNUM/1/,INITRN/100/,RUN,RODTOT/0/
      NAMELIST/INPUT/ INITRN,RUN,WALINT
C
C ARRAYS FOR ACCUMULATION OF SUMMARY STATISTICS FOR OUTPUT AT END OF
C SIMULATION.
C
C
      INTEGER IAN1(12),IAN3(12),IAN5(12),IAN7(12),IAN8(12),
     X    IAN9(12),IAN10(12)
      REAL AN2(12),AN4(12),AN6(12),AN11(75),AN12(75),AN13(75)
C
C.....................................................................
C
C READ IN ITEM DATA.
C
      READ(4,500) (SERIAL(I),PRICE(I),RECOH(I),ACTOH(I),DUES(I),
     X    Q(I),RP(I),DBAR(I),LT(I),MAD(I),I=1,561)
500   FORMAT(I3,T22,F5.2,T28,I4,T33,I4,T38,I3,T42,I4,T47,I4,T52,
     X    F6.2,T59,F2.1,T62,F5.2)
      WRITE(8,503)
503   FORMAT('1....  SIMULATION OUTPUT .....................')
C
C OPTIONAL OUTPUT FOR PRINTING ITEM DATA WITH HEADINGS.
C
C     WRITE(8,502)
502   FORMAT(/T5,'SERIAL',T16,'PRICE',T26,'RECOH',T36,'ACTOH',
     X    T47,'DUES',T60,'Q',T69,'RP',T77,'DBAR',T89,'LT',T108,'MAD'/)
C     WRITE(8,501) (SERIAL(I),PRICE(I),RECOH(I),ACTOH(I),DUES(I),
C    X    Q(I),RP(I),DBAR(I),LT(I),MAD(I),I=1,561)
501   FORMAT(T7,I4,T15,F6.2,T26,I5,T36,I5,T47,I4,T56,I5,T66,
     X    I5,T74,F7.2,T88,F3.1,T105,F6.2)
C     WRITE(8,502)
C
C.....................................................................
C
C
C QUERY USER VIA TERMINAL FOR SIMULATION PARAMETERS.  PROGRAM CAN BE
C MODIFIED FOR BATCH PROCESSING BY SUPPRESSING QUESTIONS.  ALL
C NUMERICAL RESPONSES MUST BE IN REAL FORM.
C
      WRITE(6,6001)
6001  FORMAT(' PRESENT PARAMETERS ARE:')
      WRITE(6,INPUT)
      WRITE(6,6002)
6002  FORMAT(' ANY PARAMETER CHANGES?')
      READ(5,6003) ANS
6003  FORMAT(A1)
```

```
      IF(ANS.EQ.Y) READ (5,INPUT)
150   WRITE(6,600)
600   FORMAT(' DURATION IN YEARS OR DAYS?')
      READ(5,602)ANS
602   FORMAT(A1)
      IF(.NOT.(ANS.EQ.Y.OR.ANS.EQ.D)) GO TO 150
601   FORMAT(G10.0)
      WRITE(6,603)
603   FORMAT(' DURATION?')
      READ(5,601)DURATN
      IF(ANS.EQ.Y)FINISH=DURATN*364
      IF(ANS.EQ.D)FINISH=DURATN
      WRITE(8,605) FINISH
605   FORMAT(////' FINISH=',I6,'   DAYS'///)
C
C QUERY USER VIA TERMINAL FOR INVENTORY OPTION
C
C USER TYPES: 'WALL' FOR WALL TO WALL INVENTORY
C           : 'PBUY' FOR PRIOR TO BUY INVENTORY
C
5050  WRITE(6,505)
505   FORMAT(' INVENTORY OPTION: WALL OR PBUY?')
      READ(5,510) INVOP
510   FORMAT(A4)
      IF(.NOT.(INVOP.EQ.WALL.OR.INVOP.EQ.PBUY))
     X   GO TO 5050
      IF(INVOP.EQ.WALL)WRITE(6,5100)
5100  FORMAT(' INVENTORY INTERVAL IN DAYS?')
      IF(INVOP.EQ.WALL)READ(5,5101)WALINT
5101  FORMAT(G10.0)
      INVINT=WALINT
      WRITE(8,INPUT)
C
C.....
C
C INITIALIZE RANDOM NUMBER GENERATORS WITH INITRN.
C INITRN IS NUMBER OF TIMES GENERATORS ARE CYCLED PRIOR TO GENERATING
C THE FIRST RANDOM NUMBER FOR USE BY THE SIMULATION.
C
      RN=RNERR(0,RUN)
      RN=RNERR(INITRN,RUN)
      RN=RNDMD(0)
      RN=RNDMD(INITRN)
      RN=RNNORM(0)
      RN=RNNORM(INITRN)
C
C.....
C
```

```
C GENERATE DUE DATES FOR THOSE DUES OUTSTANDING ON DAY 1.  THE
C ASSUMPTION IS THAT ALL DUES OUTSTANDING AT BEGINNING OF SIMULATION
C WILL ARRIVE UNIFORMLY DURING THE FIRST 60 DAYS.
C
      J=1
      DO 1 I=1,561
      IF(DUES(I).EQ.0) GO TO 1
      ORDQN(J)=DUES(I)
      INDEX(J)=I
      RN=RNNORM(1)
      DUEDAT(J)=RN*60
      IF(DUEDAT(J).EQ.0)DUEDAT(J)=1
      J=J+1
1     CONTINUE
C
C.....................................................................
C
C COMPUTE P AND NU FOR ALL RECORDS, AND NEW REORDER POINTS(IF DESIRED).
C
      DO 2 I=1,561
      VAR=(MAD(I)/.8)**2/91.
C
C OPEN SUBROUTINE FOR RESETTING REORDER POINTS TO PROVIDE
C 85 PER CENT PROTECTION FOR ALL ITEMS.  DELETE FOR NORMAL RUNS BY
C PUTTING A 'C' IN COLUMN 1.
C RMU IS MEAN LEAD TIME DEMAND.
C RECOH IS SET SO THAT THE INVENTORY POSITION IS RANDOM BETWEEN RP AND
C (RP+Q).  RECORD ACCURACY IS SET AT 100 PER CENT.
C
C     SIGMA=SQRT((MAD(I)/.8)**2*LT(I))
C     RMU=DBAR(I)*LT(I)
C     RP(I)=RMU+1.04*SIGMA+.5
C     RECOH(I)=RP(I)+Q(I)*RNNORM(1)-DUES(I)
C     IF(RECOH(I).LT.0) RECOH(I)=0
C     ACTOH(I)=RECOH(I)
C
C END OF OPEN SUBROUTINE.
C
      P(I)=(2.*DBAR(I)/91.)/(VAR+DBAR(I)/91.)
      NU(I)=P(I)*(DBAR(I)/91.)
      IF(P(I).GE.1.)P(I)=.9999
2     CONTINUE
C
C.....................................................................
C
C COMPUTE SUBMISSION DATES OF FIRST REQUISITIONS AND MAKE DAY 1 BUYS.
C FOR A CLEAN PBUY RUN, INITIAL RECORD ACCURACY IS SET AT 100
C PER CENT.
```

```
C
      IF(RUN.EQ.0.AND.INVOP.EQ.PBUY)CALL CLEAN(RECOH,ACTOH)
      DO 3 I=1,561
      NXTREQ(I)=TODAY-ALOG(RNDMD(1))/NU(I)
3     CALL BUY(I,J)
C
C.....................................................................
C
C DUMP P AND NU FOR ITEM SAMPLE IF DESIRED.
C
C     WRITE(8,22222)NXTREQ
22222 FORMAT(10I10)
C     WRITE(8,22221)P
22221 FORMAT(10F12.7)
C     WRITE(8,22221)NU
C
C.....................................................................
C
C BEGIN A DAY'S ROUTINE WITH RECEIPT PROCESSING
C
C INITIALIZE GRAND LOOP FOR TOTAL LENGTH OF SIMULATION
C
      DO 1000 TODAY=1,FINISH
C
C PROCESS RECEIPTS
C
C SET N=J.  N IS TOTAL NUMBER OF BUYS MADE TO DATE.  WILL INCREMENT
C J IN LOOP IF BUY OCCURS.
C 77 IS DROPOUT FOR RECT PROCESS AND BO RELEASE FOR ANY GIVEN RECT
C
      N=J
      DO 77 K=1,N
      IF(DUEDAT(K).NE.TODAY) GO TO 77
      RN=RNERR(1,RUN)
      I=INDEX(K)
      IF(RN.GT..9599) GO TO 70
C
C PROCESS RECEIPT CORRECTLY: RECTOK
C
      RECTOK=RECTOK+1
      RECOH(I)=RECOH(I)+ORDQN(K)
      DUES(I)=DUES(I)-ORDQN(K)
      IF(DUES(I).LT.0) DUES(I)=0
      ACTOH(I)=ACTOH(I)+ORDQN(K)
      GO TO 76
70    IF(RN.GT..9736) GO TO 71
C
C QUANTITY ERROR OVER:ORDQN<QTY ACTUALLY RECEIVED: RECTEO
```

```
C
      RECTEO=RECTEO+1
      RECOH(I)=RECOH(I)+ORDQN(K)
      DUES(I)=DUES(I)-ORDQN(K)
      IF(DUES(I).LT.0) DUES(I)=0
      ACTOH(I)=ACTOH(I)+1.08*ORDQN(K)+.5
      GO TO 76
71    IF(RN.GT..9874) GO TO 72
C
C QTY ERROR UNDER: ORDQN>QTY ACTUALLY RECEIVED: RECTEU
C
      RECTEU=RECTEU+1
      RECOH(I)=RECOH(I)+ORDQN(K)
      DUES(I)=DUES(I)-ORDQN(K)
      IF(DUES(I).LT.0) DUES(I)=0
      ACTOH(I)=ACTOH(I)+.94*ORDQN(K)+.5
      GO TO 76
72    IF(RN.GT..9938) GO TO 73
C
C QTY NOT POSTED TO RECOH OR DUES: RECTNP
C
      RECTNP=RECTNP+1
      ACTOH(I)=ACTOH(I)+ORDQN(K)
      GO TO 76
C
C QTY POSTED TO WRONG STOCK RECORD
C
73    RECTPW=RECTPW+1
      ACTOH(I)=ACTOH(I)+ORDQN(K)
      CALL NUREC(PRICE,I,NEWREC)
      RECOH(NEWREC)=RECOH(NEWREC)+ORDQN(K)
      DUES(NEWREC)=DUES(NEWREC)-ORDQN(K)
      IF(DUES(NEWREC).LT.0) DUES(NEWREC)=0
C
C.....................................................................
C
C BO RELEASE ROUTINE
C
C 77 IS DROPOUT FOR RECEIPT PROCESS AND BO RELEASE FOR ANY GIVEN RECEIPT
C
76    IF(RECOH(I).EQ.0) GO TO 77
      IF(BO(I).EQ.0) GO TO 77
C
      ABOREL=ABOREL+1
      IF(RECOH(I).LT.BO(I)) GO TO 80
      ISSQN=BO(I)
      BO(I)=0
      RECOH(I)=RECOH(I)-ISSQN
```

```
      GO TO 81
C
80    ISSQN=RECOH(I)
      BO(I)=BO(I)-RECOH(I)
      RECOH(I)=0
C
81    IF(ISSQN.LE.ACTOH(I)) GO TO 82
      REFUSL=REFUSL+1
C
      IF(ACTOH(I).EQ.0) GO TO 810
      BO(I)=BO(I)+ISSQN-ACTOH(I)
      ACTOH(I)=0
      BOREL=BOREL+1
      GO TO 811
C
810   BO(I)=BO(I)+ISSQN
811   CALL SPOT(RECOH,ACTOH,I,RUN)
      CALL BUY(I,J)
      GO TO 77
C
82    CALL ISBOER(ISSQN,RECOH,ACTOH,PRICE, I,RUN)
      BOREL=BOREL+1
      CALL BUY(I,J)
77    CONTINUE
C
C.....................................................................
C
C REQUISITION PROCESSING ROUTINE
C
      DO 101 I=1,561
      IF(NXTREQ(I).NE.TODAY) GO TO 101
99     RN=RNDMD(1)
90    REQSIZ=ALOG(RN)/ALOG(1.-P(I))+.5
      IF(REQSIZ.EQ.0)REQSIZ=1
      IF(REQSIZ.GT.10000) GO TO 90
      CUMREQ=CUMREQ+1
      CUMDMD(I)=CUMDMD(I)+REQSIZ
C
C COMPARE REQSIZ TO RECOH
C
      IF(REQSIZ.LE.RECOH(I)) GO TO 91
      ISSQN=RECOH(I)
      RECOH(I)=0
      BO(I)=BO(I)+REQSIZ-ISSQN
      APARFL=APARFL+1
      IF(ISSQN.GT.ACTOH(I)) GO TO 92
      IFLAG=0
      CALL ISSERR(ISSQN,RECOH,ACTOH,PRICE,I,IFLAG,COMPFL,PARTFL,RUN)
```

```
      GO TO 100
C
91    RECOH(I)=RECOH(I)-REQSIZ
      ISSQN=REQSIZ
      ACOMFL=ACOMFL+1
      IF(ISSQN.GT.ACTOH(I)) GO TO 92
      IFLAG=1
      CALL ISSERR(ISSQN,RECOH,ACTOH,PRICE,I,IFLAG,COMPFL,PARTFL,RUN)
      GO TO 100
C
92    REFUSL=REFUSL+1
      IF(ACTOH(I).EQ.0) GO TO 93
      PARTFL=PARTFL+1
      BO(I)=BO(I)+ISSQN-ACTOH(I)
      ACTOH(I)=0
      CALL SPOT(RECOH,ACTOH,I,RUN)
      GO TO 100
93    BO(I)=BO(I)+ISSQN
      CALL SPOT(RECOH,ACTOH,I,RUN)
100   CALL BUY(I,J)
      NXTREQ(I)=TODAY-ALOG(RNDMD(1))/NU(I)
      IF(NXTREQ(I).EQ.TODAY) GO TO 99
101   CONTINUE
C
C.....................................................................
C
C DAILY UPDATE AFTER PROCESSING ALL RECEIPTS AND ISSUES
C
      DO 112 I=1,561
112   BODAYS(I)=BODAYS(I)+BO(I)
C
C END OF DAY: CHECK FOR REPORT TIME AND/OR FINISH TIME
C
C DEBUGGING AID.  CHECKS FOR NEGATIVE RECOH, ACTOH, OR DUES FIELD.
C
C     DO 113 I=1,561
C     IF(RECOH(I).LT.0.OR.ACTOH(I).LT.0.OR.DUES(I).LT.0) WRITE(8,998)
C    X   TODAY
998   FORMAT(' NEG ON DAY',I6)
113   CONTINUE
C
C IF TODAY IS END OF QUARTER, ACCUMULATE QUARTERLY DEMAND AND BUY
C STATISTICS.
C
      IF(MOD(TODAY,91).EQ.0) CALL QTR$(QTRNUM,CUMDMD,CUMBUY,PRICE,
     X   CUMDM$,BUY$)
C
C COMPUTE DOLLAR VALUE STOCK HELD, BOTH ACTUAL AND
```

```
C   RECORDED.  COMPUTE TODAY'S RECORD ACCURACY.
C
      INVOK=0
      DO 1130  I=1,561
      RECOH$(TODAY)=RECOH$(TODAY)+RECOH(I)*PRICE(I)
      ACTOH$(TODAY)=ACTOH$(TODAY)+ACTOH(I)*PRICE(I)
1130  IF(RECOH(I).EQ.ACTOH(I))INVOK=INVOK+1
      RECACC(TODAY)=INVOK/561.
C
C PERFORM WALL-TO-WALL INVENTORY IF REQUIRED.
C
      IF(MOD(TODAY,INVINT).EQ.0)CALL WALLOP(RECOH,ACTOH,TODAY,
     X   INVINT,RECACC,RECOH$,ACTOH$,RUN,ACCBAR,RECBAR,ACTBAR,
     X   INVOP,PBUY)
      IF(MOD(TODAY,INVINT).EQ.0) GO TO 1001
C
C PRINT ANNUAL REPORT IF TODAY IS END OF YEAR OR END OF SIMULATION.
C
1131  IF(TODAY.EQ.FINISH) GO TO 900
      DO 111 NUM=1,15
      IF(TODAY.EQ.NUM*364) GO TO 900
111   CONTINUE
1000  CONTINUE
      STOP
C
C END OF GRAND LOOP AND STOP INSTRUCTION.
C
C CHECK FOR BUYS AFTER WALL TO WALL INVENTORY
C
1001  DO 1002 I=1,561
1002  CALL BUY(I,J)
C
C SAVE INVENTORY INTERVAL PERIOD STATISTICS FOR SUMMARY REPORT AT END
C OF SIMULATION.  'LASTPD' AT END OF SIMULATION CONTAINS TOTAL NUMBER
C OF INVENTORY PERIODS FOR THIS RUN.
C
      I=TODAY/INVINT
      AN11(I)=ACCBAR
      AN12(I)=RECBAR
      AN13(I)=ACTBAR
      LASTPD=I
      GO TO 1131
C
C.....................................................................
C
C REPORT GENERATOR
C
900   DO 115 I=1,561
```

```
115   BODTOT=BODTCT+BODAYS(I)
      WRITE(8,707) TODAY
707   FORMAT('1  TODAY IS',I7//)
      WRITE(8,710)
C
C CALCULATE REQUIRED PERCENTAGES.
C
      CMFLPC=CCMPFL*100./CUMREQ
      PTFLPC=PARTFL*100./CUMREQ
      ACMFPC=ACOMFL*100./CUMREQ
      APTFPC=APARFL*100./CUMREQ
      CUMRCT=RECTCK+RECTEO+RECTEU+RECTNP+RECTPW
      RCTOKP=RECTOK*100./CUMRCT
      RCTEOP=RECTEO*100./CUMRCT
      RCTEUP=RECTEU*100./CUMRCT
      RCTNPP=RECTNP*100./CUMRCT
      RCTPWP=RECTPW*100./CUMRCT
      CUMISS=ISSOK+ISSEO+ISSEU+ISSWID
      ISSOKP=ISSOK*100./CUMISS
      ISSEOP=ISSEO*100./CUMISS
      ISSEUP=ISSEU*100./CUMISS
      ISWIDP=ISSWID*100./CUMISS
710   FORMAT(//////' SUPPLY PERFORMANCE MEASURES:')
      WRITE(8,711)CUMREQ,COMPFL,CMFLPC,PARTFL,PTFLPC,ACOMFL,
     X   ACMFPC,APARFL,APTFPC,
     X   ABOREL,BOREL,BODTOT,
     X   BUYS,REFUSL
711   FORMAT(/' CUMREQ=',I10/
     X   ' COMPFL=',I10,'=',F6.2,' PER CENT OF CUMREQ'/
     X   ' PARTFL=',I10,'=',F6.2/
     X   ' ACOMFL=',I10,'=',F6.2/
     X   ' APARFL=',I10,'=',F6.2/
     X   ' ABOREL=',I10/' BOREL=',I11/' TOTAL BODAYS(BODTOT)=',I10/
     X   ' BUYS=',I12/' REFUSL=',I10)
      WRITE(8,712)
712   FORMAT(///' ERROR MEASURES:')
      WRITE(8,713)RECTOK,RCTOKP,RECTEO,RCTEOP,RECTEU,RCTEUP,
     X   RECTNP,RCTNPP,RECTPW,RCTPWP,ISSOK,ISSOKP,ISSEO,ISSEOP,
     X   ISSEU,ISSEUP,ISSWID,ISWIDP
713   FORMAT(/' RECTOK=',I10,'=',F6.2,' PER CENT OF TOTAL RECEIPTS'/
     X   ' RECTEO=',I10,'=',F6.2/
     X   ' RECTEU=',I10,'=',F6.2/' RECTNP=',I10,'=',F6.2/
     X   ' RECTPW=',I10,'=',F6.2/
     X   ' ISSOK=',I11,'=',F6.2,' PER CENT OF TOTAL ISSUES(INCLUDES BO R
     XELEASES)'/' ISSEO=',I11,'=',F6.2/' ISSEU=',I11,'=',F6.2/
     X   ' ISSWID=',I10,'=',F6.2)
C
C CALCULATE AND PRINT QUARTERLY DEMAND$ AND BUY$.
```

```
C
      WRITE(8,7130)
7130  FORMAT(//' QTR',T9,'DEMAND$',T32,'BUY$'/)
7131  FORMAT(T3,I2,T6,F10.2,T21,F15.2)
      I1=QTRNUM-4
      I4=QTRNUM-1
      DO 7132  II=I1,I4
      IF(II.EQ.1)A=CUMDM$(1)
      IF(II.EQ.1)B=BUY$(1)
      IF(II.NE.1)A=CUMDM$(II)-CUMDM$(II-1)
      IF(II.NE.1)B=BUY$(II)-BUY$(II-1)
7132  WRITE(8,7131) II,A,B
C
C DUMP BUY VECTORS IF DESIRED.
C
      WRITE(8,7133)
7133  FORMAT('1')
C     WRITE(8,709)
709   FORMAT(////T15,'SERIAL',T31,'RECOH',T46,'ACTOH',T64,'BO',
     X   T75,'CUMDMD',T90,'BODAYS',T107,'DUES',T115,'NXTREQ'/)
C     WRITE(8,714)(SERIAL(I),RECOH(I),ACTOH(I),BO(I),CUMDMD(I),
C    X   BODAYS(I),DUES(I),NXTREQ(I),I=1,561)
714   FORMAT(T5,8I15)
C     IF(TODAY.EQ.FINISH) CALL E8
C     IF(TODAY.EQ.FINISH)WRITE(8,715)
715   FORMAT(//T21,'J',T36,'INDEX',T57,'DUEDAT',T76,'ORDQN'/)
C     IF(TODAY.EQ.FINISH) WRITE(8,716)(I,INDEX(I),DUEDAT(I),
C    X   ORDQN(I),I=1,J)
716   FORMAT(T19,I3,T31,I10,T51,I10,T71,I10)
C
C SAVE ANNUAL STATISTICS FOR SUMMARY
C
      I=TODAY/364.
      IAN1(I)=COMPFL
      AN2(I)=CMFLPC
      IAN3(I)=PARTFL
      AN4(I)=PTFLPC
      IAN5(I)=CUMREQ-COMPFL-PARTFL
      AN6(I)=100.-CMFLPC-PTFLPC
      IAN7(I)=BOREL
      IAN8(I)=BODTOT
      IAN9(I)=BUYS
      IAN10(I)=REFUSL
C
C
C RE-INITIALIZE COUNTERS FOR NEXT REPORT DUEDAT
C
      DO 999 I=1,561
```

```
999    BODAYS(I)=0
       CUMREQ=0
       COMPFL=0
       PARTFL=0
       ABOREL=0
       BOREL=0
       ACOMFL=0
       APARFL=0
       REFUSL=0
       RECHL$=0.
       ACTHL$=0.
       BUYS=0
       RECTOK=0
       BODTOT=0
       RECTEO=0
       RECTEU=0
       RECTNP=0
       RECTPW=0
       ISSOK=0
       ISSEO=0
       ISSEU=0
       ISSWID=0
C
C  PRINT CUMDMD AT END OF SIMULATION IF DESIRED.
C
C      IF(TODAY.EQ.FINISH) WRITE(8,99999) CUMDMD
C
C
C  PRINT SUMMARY STATISTICS REPORT AT END UF SIMULATION.
C
       IF(TODAY.EQ.FINISH) GO TO 9991
99999  FORMAT(10I10)
C
C  SEE PROGRAMMING GUIDE FOR EXPLANATION OF E8 SUBROUTINE.
C
99910  CALL E8
       GO TO 111
C
C  OPEN SUBROUTINE: SUMMARY STATISTICS REPORT GENERATOR.
C
9991   WRITE(8,9992)INITRN,DURATN,RUN,INVINT
9992   FORMAT('1RUN ID:INITRN=',I3,T66,'PAGE 1'/
      X    T8,'YEARS=',F4.1/T8,'RUN=',I6,'  (1=DIRTY,0=CLEAN)'/
      X    T8,'WALL=',I5)
       WRITE(8,9993)
9993   FORMAT(//' S U M M A R Y   S T A T I S T I C S'//
      X    ' ANNUAL:'//' YEAR  COMPFL....  PARTFL....  DIFF....**  BOREL
      X    BODAYS*  BUYS  REFUSL'/
```

```
     X    T9,'NO.',T16,'PC',T21,'NO.',T28,'PC',T33,'NO.',T40,'PC'//)
      N=FINISH/364.
      DO 9995 I=1,N
9995  WRITE(8,9994)I,IAN1(I),AN2(I),IAN3(I),AN4(I),IAN5(I),
     X    AN6(I),IAN7(I),IAN8(I),IAN9(I),IAN10(I)
9994  FORMAT(T3,I2,T8,I4,T14,F4.1,T20,I4,T26,F4.1,T32,I4,
     X    T38,F4.1,T45,I4,T51,I10,T63,I4,T69,I3)
C
C  COMPUTE MEANS FOR ALL YEARS EXCEPT THE FIRST.
C
      M1=IXBAR(IAN1,N)
      AM2=XBAR(AN2,N)
      M3=IXBAR(IAN3,N)
      AM4=XBAR(AN4,N)
      M5=IXBAR(IAN5,N)
      AM6=XBAR(AN6,N)
      M7=IXBAR(IAN7,N)
      M8=IXBAR(IAN8,N)
      M9=IXBAR(IAN9,N)
      M10=IXBAR(IAN10,N)
      WRITE(8,9996)M1,AM2,M3,AM4,M5,AM6,M7,M8,M9,M10
9996  FORMAT(/' MEAN',T8,I4,T14,F4.1,T20,I4,T26,F4.1,T32,I4,
     X    T38,F4.1,T45,I4,T51,I10,T63,I4,T69,I3)
      WRITE(8,9997)
9997  FORMAT('1PERIODIC:',T66,'PAGE 2'//
     X    ' PERIOD  RECACC     RECOH$**     ACTOH$**')
C
      DO 99971 I=1,LASTPD
      AM11=AM11+AN11(I)
      AM12=AM12+AN12(I)
99971 AM13=AM13+AN13(I)
      WRITE(8,9998)(I,AN11(I),AN12(I),AN13(I),I=1,LASTPD)
9998  FORMAT(T6,I2,T10,F4.1,T18,F11.2,T31,F11.2)
      WRITE(8,9999)AM11,AM12,AM13
9999  FORMAT(/' SUM',T8,F7.1,T18,F11.2,T31,F11.2)
      GO TO 99910
      END
C
C  END OF MAIN PROGRAM.
C
C.....
C.....
C.....
C
C  FOLLOWING ARE SIMULATION SUBPROGRAMS.
C
C.....
C
```

```
C 'BUY' COMPUTES THE INVENTORY POSITION FOR ITEM I, AND MAKES BUY J
C IF NECESSARY.
C A SCHEDULED INVENTORY IS TAKEN IF THE PBUY OPTION WAS SELECTED.
C PROCUREMENT LEAD TIMES (DUEDAT) ARE GENERATED USING NORMAL DISTRIBU-
C TION.  ITIME IS UPPER BOUND ON DUEDAT.
C


      SUBROUTINE BUY(I,J)
      COMMON /BUYY/RECOH,DUES,BO,RP,Q,INDEX,DUEDAT,ORDQN,LT,TODAY,BUYS,
     X   CUMBUY,ACTOH,RUN,INVOP,PBUY
      REAL LT(561),INVOP,PBUY
      INTEGER RECOH(561),DUES(561),BO(561),RP*2(561),Q*2(561),
     X   DUEDAT*2(6500),ORDQN*2(6500),TODAY,BUYS,CUMBUY(561),
     X   INDEX*2(6500),ACTOH(561),RUN
      IP=RECOH(I)+DUES(I)-BO(I)
      IF(IP.GT.RP(I))RETURN
      IF(INVOP.EQ.PBUY)CALL SPOT(RECOH,ACTOH,I,RUN)
      BUYS=BUYS+1
      ORDQN(J)=((RP(I)-IP)/Q(I)+1)*Q(I)
      CUMBUY(I)=CUMBUY(I)+ORDQN(J)
      INDEX(J)=I
      DUES(I)=DUES(I)+ORDQN(J)
      TN=0
      DO 5 K=1,12
5     TN=TN+RNNORM(1)
      DUEDAT(J)=LT(I)*91.+LT(I)*91.*.29*(TN-6.0)+TODAY
      IF(DUEDAT(J).LT.TODAY) DUEDAT(J)=TODAY
      ITIME=2.33*LT(I)*91.*.29 + LT(I)*91. + TODAY
      IF(ITIME.LT.DUEDAT(J))DUEDAT(J)=ITIME
      J=J+1
      RETURN
      END
C
C.....................................................................
C
C 'NUREC' GENERATES A RANDOM ITEM NUMBER IN THE SAME PRICE CATEGORY
C AS ITEM I.  USED FOR GENERATING INCORRECT POSTINGS OF RECEIPTS
C AND INCORRECT ISSUES.
C


      SUBROUTINE NUREC(PRICE,I,NEWREC)
      REAL PRICE(1)
1     NEWREC=RNNORM(1)*561
      IF(.NOT.(((0.0.LE.PRICE(I).AND.PRICE(I).LE.1.00).AND.
     X   (0.00.LE.PRICE(NEWREC).AND.PRICE(NEWREC).LE.1.00))
     X   .OR.((1.01.LE.PRICE(I).AND.PRICE(I).LE.10.00).AND.
```

```
     X    (1.01.LE.PRICE(NEWREC).AND.PRICE(NEWREC).LE.10.00))
     X    .OR.((10.01.LE.PRICE(I).AND.PRICE(I).LE.100.00).AND.
     X    (10.01.LE.PRICE(NEWREC).AND.PRICE(NEWREC).LE.100.00))
     X    .OR.(100.01.LE.PRICE(I).AND.100.01.LE.PRICE(NEWREC)))) GO TO 1
      RETURN
      END
C
C.....................................................................
C
C 'SPOT' PERFORMS A CORRECT INVENTORY ON ITEM I 96 PER CENT OF THE
C TIME.  OTHER ERRORS ARE  GENERATED AS INDICATED.  IN ALL CASES,
C THE MAGNITUDE OF THE MAXIMUM ERROR IS 50 PER CENT OF ACTOH.
C


      SUBROUTINE SPOT(RECOH,ACTOH,I,RUN)
      INTEGER RECOH(1),ACTOH(1),RUN
      RN=RNERR(1,RUN)
      IF(RN.LE..96)RECOH(I)=ACTOH(I)
      IF(RN.LE..96)RETURN
      IF(RN.GT..96.AND.RN.LE..965)E=1.
      IF(RN.GT..965.AND.RN.LE..970)E=-1.
      IF(RN.GT..970.AND.RN.LE..975)E=2.
      IF(RN.GT..975.AND.RN.LE..980)E=-2.
      IF(RN.GT..980.AND.RN.LE..984)E=5.
      IF(RN.GT..984.AND.RN.LE..987)E=10.
      IF(RN.GT..987.AND.RN.LE..989)E=50.
      IF(RN.GT..989.AND.RN.LE..990)E=100.
      IF(RN.GT..990.AND.RN.LE..994)E=-5.
      IF(RN.GT..994.AND.RN.LE..997)E=-10.
      IF(RN.GT..997.AND.RN.LE..999)E=-50.
      IF(RN.GT..999.AND.RN.LE.1.000)E=-100.
      MIN=ACTOH(I)/2.+.5
      MAX=1.5*ACTOH(I)+.5
      IF(ACTOH(I).EQ.2)MIN=1
      IF(ACTOH(I).EQ.1)MIN=0
      IF(ACTOH(I).EQ.0)MIN=0
      ITEST=ACTOH(I)+E
      IF(ITEST.GE.MIN.AND.ITEST.LE.MAX)RECOH(I)=ITEST
      IF(ITEST.LT.MIN)RECOH(I)=MIN
      IF(ITEST.GT.MAX)RECOH(I)=MAX
      RETURN
      END
C
C.....................................................................
C
C 'ISBOER' RELEASES BO WITH ERRORS, AS  NOTED IN FLOW CHART.
C
```

```
      SUBROUTINE ISBOER(ISSQN,RECOH,ACTOH,PRICE,I,RUN)
      INTEGER RECOH(1),ACTOH(1),RUN
      REAL PRICE(1)
      COMMON /ISSUE/ ISSOK,ISSEO,ISSEU,ISSWID
      RN=RNERR(1,RUN)
C
C CHECK FOR CORRECT ISSUE: ISSOK
C
      IF(RN.GT..9774) GO TO 1
      ISSOK=ISSOK+1
      ACTOH(I)=ACTOH(I)-ISSQN
      RETURN
C
C CHECK FOR OVERISSUE: ISSEO
C
1     IF(RN.GT..9847) GO TO 2
      ISSEO=ISSEO+1
      IF(RN.LE..9822) ACTOH(I)=ACTOH(I)-1.07*ISSQN-.5
      IF(RN.GT..9822.AND.RN.LE..9840) ACTOH(I)=ACTOH(I)-1.15*ISSQN-.5
      IF(RN.GT..9840.AND.RN.LE..9847) ACTOH(I)=ACTOH(I)-1.30*ISSQN-.5
      IF(ACTOH(I).LT.0) ACTOH(I)=0
      RETURN
C
C CHECK FOR UNDERISSUE: ISSEU
C
2     IF(RN.GT..9922) GO TO 3
      ISSEU=ISSEU+1
      IF(RN.LE..9897) ACTOH(I)=ACTOH(I)-.92*ISSQN-.5
      IF(RN.GT..9897.AND.RN.LE..9909) ACTOH(I)=ACTOH(I)-.80*ISSQN-.5
      IF(RN.GT..9909.AND.RN.LE..9922) ACTOH(I)=ACTOH(I)-.50*ISSQN-.5
      RETURN
C
C ISSUE ISSQN FROM RANDOM ID ACTOH, POST RECOH PROPERLY: ISSWID
C
3     ISSWID=ISSWID+1
4     CALL NUREC(PRICE,I,NEWREC)
      IF(ISSQN.GT.ACTOH(NEWREC)) GO TO 4
      ACTOH(NEWREC)=ACTOH(NEWREC)-ISSQN
      RETURN
      END
C
C.....................................................................
C
C RANDOM NUMBER GENERATORS, USED AS FOLLOWS:
C   RNDMD: GENERATES DEMAND PATTERN.
C   RNERR: GENERATES ERRORS FOR DIRTY RUNS.
```

```
C    RNNORM: FOR ALL OTHER USES, SUCH AS PROCUREMENT LEAD TIMES FOR BUYS.
C
C N IS NUMBER OF TIMES TO CYCLE.
C RUN: 0 IMPLIES USE FOR A CLEAN RUN.
C      1 IMPLIES USE FOR A DIRTY RUN.
C


      FUNCTION RNDMD(N)
      NR1=N
      IF(NR1)11,11,21
11    IX1=30517
      NR1=NR1+1
21    DO 49 I=1,NR1
      IY1=IX1*65539
      IF(IY1)7,8,8
7     IY1=IY1+2147483647+1
8     RNDMD=IY1
      RNDMD=RNDMD*.4656613 E-9
49    IX1=IY1
      RETURN
      END
C


      FUNCTION RNERR(N,RUN)
      INTEGER RUN
      IF(RUN.EQ.0) RNERR=.5
      IF(RUN.EQ.0) RETURN
      NR1=N
      IF(NR1)11,11,21
11    IX1=30517
      NR1=NR1+1
21    DO 49 I=1,NR1
      IY1=IX1*65539
      IF(IY1)7,8,8
7     IY1=IY1+2147483647+1
8     RNERR=IY1
      RNERR=RNERR*.4656613 E-9
49    IX1=IY1
      RETURN
      END
C
```

```
      FUNCTION RNNORM(N)
      NR1=N
      IF(NR1)11,11,21
11    IX1=30517
      NR1=NR1+1
21    DO 49 I=1,NR1
      IY1=IX1*65539
      IF(IY1)7,8,8
7     IY1=IY1+2147483647+1
8     RNNORM=IY1
      RNNORM=RNNORM*.4656613 E-9
49    IX1=IY1
      RETURN
      END
C
C.....................................................................
C
C 'ISSERR' MAKES ISSUES WITH ERRORS.  SIMILAR TO ISBOER SUBROUTINE.
C


      SUBROUTINE ISSERR(ISSQN,RECOH,ACTOH,PRICE,I,IFLAG,COMPFL,PARTFL,
     X   RUN)
      INTEGER RECOH(1),ACTOH(1),COMPFL,PARTFL,RUN
      REAL PRICE(1)
      COMMON /ISSUE/ISSOK,ISSEO,ISSEU,ISSWID
      RN=RNERR(1,RUN)
C
C CHECK FOR CORRECT ISSUE: ISSOK
C
      IF(RN.GT..9774) GO TO 1
      ISSOK=ISSOK+1
      IF(IFLAG.EQ.1) COMPFL=COMPFL+1
      IF(IFLAG.EQ.0) PARTFL=PARTFL+1
      ACTOH(I)=ACTOH(I)-ISSQN
      RETURN
C
C CHECK FOR OVERISSUE: ISSEO
C
1     IF(RN.GT..9847) GO TO 2
      ISSEO=ISSEO+1
      IF(IFLAG.EQ.1) COMPFL=COMPFL+1
      IF(IFLAG.EQ.0) PARTFL=PARTFL+1
      IF(RN.LE..9822) ACTOH(I)=ACTOH(I)-1.07*ISSQN-.5
      IF(RN.GT..9822.AND.RN.LE..9840) ACTOH(I)=ACTOH(I)-1.15*ISSQN-.5
      IF(RN.GT..9840.AND.RN.LE..9847) ACTOH(I)=ACTOH(I)-1.30*ISSQN-.5
      IF(ACTOH(I).LT.0) ACTOH(I)=0
      RETURN
```

```
C
C CHECK FOR UNDERISSUE: ISSEU
C
2       IF(RN.GT..9922) GO TO 3
        ISSEU=ISSEU+1
        PARTFL=PARTFL+1
        IF(RN.LE..9897) ACTOH(I)=ACTOH(I)-.92*ISSQN-.5
        IF(RN.GT..9897.AND.RN.LE..9909) ACTOH(I)=ACTOH(I)-.80*ISSQN-.5
        IF(RN.GT..9909.AND.RN.LE..9922) ACTOH(I)=ACTOH(I)-.50*ISSQN-.5
        RETURN
C
C ISSUE ISSQN FROM RANDOM ID ACTOH, POST RECOH PROPERLY: ISSWID
C
3       ISSWID=ISSWID+1
4       CALL NUREC(PRICE,I,NEWREC)
        IF(ISSQN.GT.ACTOH(NEWREC)) GO TO 4
        ACTOH(NEWREC)=ACTOH(NEWREC)-ISSQN
        RETURN
        END
C
C.....................................................................
C
C 'QTR$' IS CALLED AT END OF EVERY QUARTER.  IT ACCUMULATES THE
C VALUES OF QUARTERLY DEMAND AND BUYS.
C


        SUBROUTINE QTR$(I,CUMDMD,CUMBUY,PRICE,CUMDM$,BUY$)
        INTEGER CUMDMD(1),CUMBUY(1)
        REAL PRICE(1),CUMDM$(1),BUY$(1)
        DO 1  J=1,561
        CUMDM$(I)=CUMDM$(I)+CUMDMD(J)*PRICE(J)
1       BUY$(I)=BUY$(I)+CUMBUY(J)*PRICE(J)
        I=I+1
        RETURN
        END
C
C.....................................................................
C
C 'WALLOP' IS THE WALL-TO-WALL INVENTORY ROUTINE, WHICH
C PERFORMS A PERICDIC WALL-TO-WALL INVENTORY, AND COMPUTES STATISTICS
C FOR INVENTORY REPORT AND INVENTORY INTERVAL.  ALSO PERFORMS
C ANNUAL STATISTICAL COMPUTATIONS FOR THE PBUY OPTION.
C
C
```

```
      SUBROUTINE WALLOP(RECOH,ACTOH,TODAY,INVINT,RECACC,RECOH$,ACTOH$,
     X   RUN,ACCBAR,RECBAR,ACTBAR,INVOP,PBUY)
      REAL RECACC(1),RECOH$(1),ACTOH$(1),INVOP,PBUY
      INTEGER ERRQN,RECOH(1),ACTOH(1),TODAY,RUN
C
C CALCULATE REPORT STATISTICS.
C
      SUM1=0.
      SUM2=0.
      SUM3=0.
      INIT=TODAY-INVINT+1
      DO 10  I=INIT,TODAY
      SUM1=SUM1+RECOH$(I)
      SUM2=SUM2+ACTOH$(I)
10    SUM3=SUM3+RECACC(I)
      RECBAR=SUM1/INVINT
      ACTBAR=SUM2/INVINT
      ACCBAR=SUM3/INVINT*100.
      RECMIN=RECOH$(INIT)
      ACTMIN=ACTOH$(INIT)
      ACCMIN=RECACC(INIT)
      J=INIT+1
      DO 11  I=J,TODAY
      RECMIN=AMIN1(RECMIN,RECOH$(I))
      ACTMIN=AMIN1(ACTMIN,ACTOH$(I))
11    ACCMIN=AMIN1(ACCMIN,RECACC(I))
      ACCMIN=ACCMIN*100.
      INVOK=0
      DO 1  I=1,561
1     IF(RECOH(I).EQ.ACTOH(I))INVOK=INVOK+1
C
C IF INVOP IS PBUY, SKIP WALL-TO-WALL INVENTORY.
C
      IF(INVOP.EQ.PBUY)GO TO 3
C
C PERFORM INVENTORY WITH ERRORS.
C
      DO 3  I=1,561
      RN=RNERR(1,RUN)
      IF(RN.LE..929)GO TO 2
      IF(ACTOH(I).GE.0.AND.ACTOH(I).LE.10)ERRQN=NORMAL(0.,1.87)
      IF(ACTOH(I).GT.10.AND.ACTOH(I).LE.20)ERRQN=NORMAL(0.,4.65)
      IF(ACTOH(I).GT.20.AND.ACTOH(I).LE.100)ERRQN=NORMAL(0.,2.20)
      IF(ACTOH(I).GT.100)ERRQN=NORMAL(0.,11.80)
      RECOH(I)=ACTOH(I)+ERRQN
      IF(RECOH(I).LT.0) RECOH(I)=0
      GO TO 3
2     RECOH(I)=ACTOH(I)
```

```
3       CONTINUE
C
        PERCNT=INVCK/561.*100.
C
C PRINT INVENTORY REPORT.
C
        WRITE(8,100)TODAY,PERCNT,ACCBAR,ACCMIN,RECBAR,RECMIN,ACTBAR,ACTMIN
100     FORMAT(///I6,'=DAY WALL-TO-WALL INVENTORY HELD'/F12.2,
     X    '=PER CENT RECORDS ACCURATE JUST PRIOR TO INVENTORY'/
     X    F12.2,'=MEAN PER CENT RECORD ACCURACY DURING PERIOD'/
     X    F12.2,'=MINIMUM PER CENT RECORD ACCURACY DURING PERIOD'/
     X    F12.2,'=MEAN DOLLAR VALUE OF RECOH DURING PERIOD'/
     X    F12.2,'=MINIMUM DOLLAR VALUE OF RECOH DURING PERIOD'/
     X    F12.2,'=MEAN DOLLAR VALUE OF ACTOH DURING PERIOD'/
     X    F12.2,'=MINIMUM DOLLAR VALUE OF ACTOH DURING PERIOD')
        INVOK=0
        DO 4  I=1,561
4       IF(RECOH(I).EQ.ACTOH(I))INVOK=INVOK+1
        PERCNT=INVOK/561.*100.
        WRITE(8,101)PERCNT
101     FORMAT(F12.2,'=PER CENT RECORDS ACCURATE JUST AFTER INVENTORY'///)
        RETURN
        END
C
C.....................................................................
C
C ROUTINE FOR GENERATING NORMAL VARIATE WITH MEAN MU, STANDARD DEVIATION
C SIGMA.
C GENERATES INTEGER NORMAL VARIATES WITH A MAGNITUDE OF AT LEAST 1.
C USED IN ERROR QUANTITY GENERATION.
C


        FUNCTION NORMAL(MU,SIGMA)
        REAL MU
        VAR=0.
        DO 1  I=1,12
        RN=RNNORM(1)
1       VAR=VAR+RN
        VAR1=(VAR-6.0)*SIGMA+MU
        IF(VAR1.GE.0..AND.VAR1.LT.1.0)VAR1=VAR1+1.0
        IF(VAR1.LT.0..AND.VAR1.GT.-1.0)VAR1=VAR1-1.0
        NORMAL=VAR1
        RETURN
        END
C
C.....................................................................
C
```

```
C THESE TWO FUNCTIONS USED FOR COMPUTING MEANS OF STATISTICS FOR ALL
C BUT FIRST YEAR OF SIMULATION.  FIRST VALUE IN VECTOR TO BE
C AVERAGED IS NOT USED.  'IXBAR' RETURNS A ROUNDED INTEGER MEAN.
C

      FUNCTION XBAR(X,N)
      REAL X(1)
      SUM=0.
      DO 1 I=2,N
1     SUM=SUM+X(I)
      XBAR=SUM/(N-1)
      RETURN
      END
C

      FUNCTION IXBAR(M,N)
      INTEGER M(1)
      SUM=0.
      DO 1 I=2,N
1     SUM=SUM+M(I)
      IXBAR=SUM/(N-1)+.5
      RETURN
      END
C
C.....................................................................
C
C ROUTINE TO PERFORM CLEAN WALL-TO-WALL INVENTORY.  USED IN CONJUNCTION
C WITH CLEAN PBUY RUN.
C

      SUBROUTINE CLEAN(RECOH,ACTOH)
      INTEGER RECOH(1),ACTOH(1)
      DO 1 I=1,561
1     RECOH(I)=ACTOH(I)
      RETURN
      END
C
C.....................................................................
```

7. ITEM DATA EMPLOYED IN THE ORIGINAL STUDY

The item characteristics (Q, MAD, etc) are defined in Section 2.2 of this report.

SERIAL	PRICE	RECOH	ACTOH	DUES	Q	RP	DBAR	LT	MAD
1	1.16	0	0	9	6	3	1.64	2.3	1.64
2	0.46	21	21	0	17	9	4.16	2.3	4.42
3	0.73	4	4	0	3	1	0.80	1.4	1.00
4	7.60	128	129	66	26	158	72.88	2.3	113.84
5	15.90	3	3	0	2	2	1.20	1.4	1.50
6	64.69	1	1	0	1	0	0.27	2.5	0.46
7	7.57	3	15	1	9	7	5.27	2.3	3.18
8	0.03	294	294	0	240	127	60.00	1.7	72.00
9	4.90	2	0	6	6	4	1.44	2.2	2.28
10	2.40	8	8	0	5	4	1.28	2.2	2.08
11	1.58	2	2	0	3	1	0.81	1.3	0.60
12	23.50	3	3	0	2	1	0.80	1.2	1.00
13	9.30	5	5	0	3	2	0.81	1.7	1.31
14	6.59	0	0	0	1	0	0.32	1.7	0.52
15	3.20	0	0	0	41	103	61.96	2.9	36.24
16	2.88	0	50	0	22	32	13.44	2.0	21.12
17	0.38	233	233	0	144	143	71.76	2.4	71.76
18	0.07	880	880	0	745	683	279.52	2.4	341.00
19	0.28	509	509	0	253	407	152.00	2.4	238.00
20	0.06	222	457	0	417	210	104.16	2.4	84.68
21	4.26	15	15	0	8	6	1.92	2.4	3.12
22	1.44	7	7	0	5	2	1.28	2.4	1.28

SERIAL	PRICE	RECOH	ACTOH	DUES	Q	RP	DBAR	LT	MAD
23	0.14	30	31	0	34	17	8.47	2.4	6.84
24	0.37	6	6	0	8	6	1.92	2.4	3.12
25	1.29	9	9	0	9	5	2.13	2.4	2.75
26	3.20	27	27	0	18	18	10.60	2.4	8.56
27	0.12	32	32	0	39	18	9.80	2.4	7.24
28	0.64	14	14	0	18	10	4.48	2.4	4.48
29	26.30	7	7	0	4	5	3.84	2.0	3.12
30	2.24	18	18	0	12	6	2.88	2.0	3.12
31	0.88	14	14	0	13	6	3.15	2.0	3.07
32	0.87	20	20	0	17	7	4.27	2.0	3.22
33	0.09	191	181	0	442	186	165.67	2.0	60.23
34	313.00	1	1	1	1	1	0.80	2.0	1.24
35	1.90	4	4	0	3	1	0.80	1.3	1.00
36	0.02	0	0	0	30	12	7.50	1.2	7.50
37	1.10	33	33	0	39	25	14.80	1.3	17.80
38	0.01	585	585	0	347	322	208.03	2.3	136.08
39	0.40	272	272	0	123	162	61.28	2.3	96.28
40	0.06	165	165	0	171	107	42.68	2.3	49.75
41	0.51	121	96	0	88	119	44.65	2.3	71.08
42	0.17	0	0	0	1	0	0.27	2.5	0.46
43	0.04	5	5	0	8	3	2.00	1.2	2.00
44	5.60	3	3	0	2	1	0.40	2.5	0.50
45	1.20	1	1	0	2	1	0.40	2.5	0.80

SERIAL	PRICE	RECOH	ACTOH	DUES	Q	RP	DBAR	LT	MAD
46	3.10	0	1	0	2	1	0.54	2.5	0.93
47	4.80	7	7	0	6	3	1.60	2.3	1.60
48	0.21	18	15	0	14	8	3.44	2.3	3.60
49	0.30	0	0	0	4	4	1.01	2.5	2.00
50	0.01	4	4	0	4	2	1.00	1.2	1.00
51	0.53	67	66	0	35	32	8.65	2.3	17.32
52	0.52	17	17	0	12	9	3.04	2.3	4.76
53	0.02	8	8	0	5	4	1.28	2.3	2.08
54	2.10	4	4	0	4	1	0.96	2.3	0.81
55	0.12	8	8	0	6	3	1.45	2.3	1.51
56	0.01	672	672	0	380	344	94.94	2.5	166.14
57	10.03	34	34	0	57	34	14.24	3.2	15.88
58	0.05	10	11	0	8	5	2.00	1.3	4.00
59	0.52	114	114	0	77	45	19.20	1.9	24.00
60	0.33	8	8	18	15	11	3.84	2.3	5.74
61	0.20	2	2	0	3	2	0.80	2.3	1.24
62	18.50	5	5	0	4	5	2.24	2.3	3.52
63	0.01	3	0	0	4	1	0.90	2.3	0.86
64	0.50	19	19	0	10	8	2.40	2.5	4.20
65	1.10	109	109	33	68	98	39.69	2.3	56.34
66	4.70	0	0	0	1	0	0.25	2.7	0.37
67	1.00	85	85	93	93	87	70.09	2.3	28.55
68	0.34	92	92	0	102	82	50.80	1.9	45.08

SERIAL	PRICE	RECOH	ACTOH	DUES	Q	RP	DBAR	LT	MAD
69	0.38	0	5	0	4	2	1.00	1.2	1.33
70	0.32	513	513	0	173	346	104.00	3.0	178.00
71	0.31	599	599	0	199	459	114.68	3.0	249.52
72	0.15	0	0	150	148	87	37.03	3.0	29.97
73	0.40	0	3	100	35	113	70.99	3.0	42.17
74	2.22	22	22	0	15	11	5.71	2.3	5.12
75	0.94	6	6	0	5	2	1.20	1.1	1.50
76	5.50	0	0	6	5	2	1.20	1.1	1.50
77	1.10	2	3	0	2	1	0.40	2.0	0.80
78	2.00	9	9	0	7	2	1.81	1.8	1.31
79	2.30	4	4	0	5	1	1.28	1.8	0.74
80	1.30	1	1	0	1	0	0.25	2.2	0.37
81	0.04	139	138	0	102	47	25.55	1.8	24.40
82	0.10	2	2	0	3	1	0.81	1.8	0.94
83	0.10	42	42	0	37	14	9.20	1.8	7.08
84	0.11	13	13	0	11	6	2.80	2.0	3.50
85	0.10	11	11	0	8	3	1.92	1.8	1.60
86	0.20	1	1	9	7	3	1.65	1.8	1.56
87	1.30	4	4	0	5	2	1.16	1.8	1.16
88	1.80	2	2	0	2	0	0.41	1.8	0.50
89	0.07	3	3	0	4	1	0.91	1.8	0.86
90	4.60	2	2	0	2	0	0.52	1.8	0.52
91	0.78	3	3	0	2	1	0.61	1.8	0.70

SERIAL	PRICE	RECOH	ACTOH	DUES	Q	RP	DBAR	LT	MAD
92	0.50	3	3	0	3	1	0.82	1.8	1.02
93	1.90	0	0	2	2	1	0.49	1.8	0.79
94	0.29	26	26	0	39	14	9.77	1.8	6.63
95	1.00	0	0	0	1	0	0.32	1.8	0.52
96	0.68	1	1	0	2	0	0.52	1.8	0.52
97	0.94	0	0	0	4	1	1.00	0.6	1.00
98	1.00	2	2	0	1	0	0.32	1.8	0.52
99	0.87	7	7	0	5	2	1.28	1.8	1.52
100	0.07	6	6	0	3	2	0.80	1.8	1.24
101	0.57	0	0	4	3	1	0.64	1.8	1.04
102	1.50	16	16	0	16	6	4.00	1.1	5.00
103	2.80	2	2	0	2	0	0.52	1.8	0.52
104	0.17	8	8	0	7	3	1.88	1.8	1.47
105	0.90	2	2	0	2	1	0.58	1.8	1.04
106	0.08	3659	3666	0	2135	2813	3202.80	1.8	980.44
107	0.33	27	27	0	20	8	4.97	1.8	4.32
108	0.09	7	5	0	28	21	6.94	2.0	12.14
109	11.60	14	14	0	7	10	4.16	1.8	7.36
110	7.20	3	3	0	3	1	0.80	2.0	1.00
111	1.80	1	1	0	1	0	0.25	2.2	0.37
112	6.30	4	4	0	4	2	1.00	1.8	1.44
113	1.00	1	1	0	1	0	0.32	1.8	0.52
114	46.00	1	1	0	2	1	1.20	1.2	1.50

SERIAL	PRICE	RECOH	ACTOH	DUES	Q	RP	DBAR	LT	MAD
115	2.80	1	1	0	1	0	0.32	1.8	0.52
116	3.10	3	3	0	2	1	1.00	1.0	1.00
117	2.30	4	0	0	3	2	0.80	1.8	1.24
118	2.60	4	4	0	9	4	2.22	1.8	2.07
119	11.10	2	2	0	1	2	1.50	0.1	1.50
120	0.65	24	25	57	54	27	20.29	1.8	13.38
121	1.50	4	2	0	6	1	1.45	1.8	0.86
122	1.33	0	0	0	1	0	0.32	1.8	0.52
123	0.28	3	3	0	5	2	1.20	1.1	1.50
124	8.60	3	3	0	4	1	1.40	1.1	1.20
125	1.00	0	0	0	5	4	1.20	2.0	2.40
126	0.48	9	9	0	6	3	1.59	2.0	1.63
127	0.29	0	1	0	10	2	2.45	1.8	1.14
128	1.00	0	0	0	4	2	0.96	1.8	1.12
129	0.41	4	4	0	3	2	0.80	1.8	1.24
130	0.63	3	3	0	3	1	0.64	1.8	1.04
131	0.41	7	7	0	5	2	1.20	1.1	1.50
132	1.00	91	91	0	59	82	29.60	1.9	56.68
133	0.08	0	0	0	2	1	0.53	2.1	0.93
134	0.19	6	6	0	5	3	1.28	1.8	2.08
135	9.07	1	1	0	1	0	0.25	2.2	0.37
136	1.04	3	3	0	3	1	0.72	1.8	0.96
137	4.10	7	7	0	6	2	1.60	1.1	2.00

SERIAL	PRICE	RECOH	ACTOH	DUES	Q	RP	DBAR	LT	MAD
138	17.20	0	0	0	1	0	0.32	1.8	0.52
139	6.60	5	5	0	3	2	0.80	1.8	1.24
140	2.00	2	2	0	2	0	0.41	1.8	0.50
141	2.60	4	4	0	3	2	0.80	1.8	1.24
142	14.60	4	4	0	4	2	1.40	1.2	1.70
143	25.80	1	1	0	1	0	0.27	2.0	0.46
144	10.00	22	17	0	6	16	8.80	2.0	11.00
145	6.30	1	1	0	1	0	0.32	1.8	0.52
146	37.80	0	0	1	1	0	0.37	1.8	0.48
147	31.90	2	2	0	2	2	1.20	2.0	1.80
148	3.90	3	3	0	5	2	1.20	1.2	1.50
149	41.70	1	1	0	1	0	0.25	2.2	0.37
150	25.80	2	2	0	2	1	1.20	1.1	1.50
151	0.50	4	4	0	3	2	0.80	1.8	1.24
152	8.10	1	1	0	1	0	0.32	1.8	0.52
153	10.00	121	121	0	30	94	90.00	1.1	112.50
154	3.00	2	2	0	3	1	0.64	1.8	1.04
155	22.60	0	0	18	7	11	10.19	1.8	6.19
156	1.00	555	556	0	245	470	508.53	1.8	211.76
157	1.40	4	4	0	4	2	0.96	1.8	1.56
158	2.00	10	10	0	18	7	4.40	1.1	5.30
159	0.16	128	38	0	180	111	67.32	1.9	54.36
160	2.50	9	9	0	7	4	1.68	1.9	2.32

SERIAL	PRICE	RECOH	ACTOH	DUES	Q	RP	DBAR	LT	MAD
161	7.90	13	13	11	12	14	9.35	1.9	7.47
162	0.89	6	6	0	8	4	2.08	1.9	2.28
163	0.10	5	5	0	4	1	1.07	2.1	0.73
164	1.20	281	280	0	143	238	214.32	2.1	90.32
165	8.51	155	174	0	84	109	63.09	2.1	58.60
166	0.32	147	147	0	102	77	38.40	1.7	48.00
167	0.50	141	141	0	93	56	34.88	2.1	25.68
168	1.95	138	138	0	62	77	74.03	2.1	25.99
169	0.32	37	40	0	144	35	36.00	1.1	20.00
170	4.00	58	58	0	34	41	40.61	1.1	34.34
171	4.75	6	6	0	6	3	1.60	1.1	2.79
172	4.50	94	94	0	40	74	79.80	1.1	69.20
173	6.88	9	9	0	6	4	2.24	1.5	2.92
174	19.40	7	7	0	4	7	3.04	2.1	4.76
175	25.00	3	3	0	3	2	1.72	2.1	1.28
176	5.15	6	6	0	8	4	2.02	2.0	2.56
177	5.95	5	5	0	5	3	1.80	2.4	1.76
178	2.70	0	0	0	6	4	1.48	2.5	2.28
179	2.60	13	13	0	15	7	3.84	2.5	2.56
180	1.10	16	16	12	33	22	8.32	2.5	9.96
181	8.18	29	28	0	11	17	8.00	1.6	13.00
182	4.20	35	35	0	30	23	30.41	1.2	13.40
183	3.50	6	6	0	6	2	1.60	1.0	2.00

SERIAL	PRICE	RECOH	ACTOH	DUES	Q	RP	DBAR	LT	MAD
184	0.80	0	0	50	41	9	15.39	1.1	3.86
185	1.90	62	62	0	24	32	11.84	2.4	19.24
186	0.69	114	10	135	83	106	41.60	2.4	60.80
187	5.80	19	19	0	12	12	6.99	2.4	5.63

REFERENCES

(1) Schrady, D. A. and W. D. Free, "Analysis of Inventory Record Accuracy," NPS55SoFs0071A, Naval Postgraduate School, Monterey, California, July 1970.

(2) Emma, C. K., "Physical Inventory Procedures at Navy Stock Points - Observations on Physical Inventory and Stock Record Accuracy," Naval Supply Research and Development Facility, Bayonne, June 1966. (Available from DDC, AD488-3171).

(3) Fleet Material Support Office, "ALRAND Working Memorandum 137," Code 97, Mechanicsburg, Pennsylvania, 6 March 1968.

INITIAL DISTRIBUTION LIST

	No. Copies
Defense Documentation Center (DDC) Cameron Station Alexandria, Virginia 22314 Attn: IRS	20
Library Naval Postgraduate School Monterey, California 93940	2
Library (SUP0833C) Naval Supply Systems Command Washington, C. C. 20390	1
Library (Code 55) Naval Postgraduate School Monterey, California 93940	1
CDR Lee Brown, SC, USN (Code 97) Fleet Material Support Office Mechanicsburg, Pennsylvania 17055	1
LCDR John M. Cook, SC, USN Code 04511F Naval Supply Systems Command Washington, D. C. 20390	1
Mr. Glenn Crum (Code 97) Fleet Material Support Office Mechanicsburg, Pennsylvania 17055	1
LCDR Paul A. Tully, SC, USN Fleet Material Support Office (Code 97) Mechanisburg, Pennsylvania 17055	1
CAPT Thomas J. Ingram, III, SC, USN Director, Research and Development Division (SUP 063) Naval Supply Systems Command Washington, D.C. 20390	5
Dean C. E. Menneken Dean of Research Administration Code 023 Naval Postgraduate School Monterey, California 93940	2

	No. Copies
Mr. James W. Prichard SUP 0613 Naval Supply Systems Command Washington, D. C. 20390	1
Mr. B. B. Rosenman Chief, AMC Inventory Research Office Frankford Arsenal Philadelphia, Pennsylvania 19137 Attn: SMUFA-W5000	1
Dr. David A. Schrady Associate Professor Department of Operations Analysis (Code 55So) Naval Postgraduate School Monterey, California 93940	5
LT W. Dean Free, SC, USN Department of Operations Analysis (Code 55Fs) Naval Postgraduate School Monterey, California 93940	40

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

Inventory Policy Evaluation Simulation

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Technical Report

5. AUTHOR(S) (First name, middle initial, last name)

W. Dean Free
David A. Schrady

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
September 1970	73	3

8a. CONTRACT OR GRANT NO. b. PROJECT NO. Work Request No. WR-0-5037 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) NPS55FsSo0091A 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Research and Development Division Naval Supply Systems Command

13. ABSTRACT

This report is a companion to the report, "Analysis of Inventory Record Accuracy" (NPS55SoFs0071A), and supplements that report by providing detailed information about the simulation model employed. The model is described, a programming guide is provided, and the program and its flow charts are given.

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Inventory record accuracy						
Supply effectiveness						
Wall-to-wall physical inventory						
Simulation						
Physical inventory						
Inventory policy						